Next-Generation Cybersecurity Threat Detection: Integration with Artificial Intelligence

Kaiwen Zheng *

School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China

* Corresponding Author Email: zhengkaiwen@bupt.edu.cn

Abstract. With the rapid advancement of information technology, cybersecurity threats have evolved into a significant challenge in modern society. Traditional detection methods often fail to address new and complex threats, especially large-scale and persistent intrusions. Artificial Intelligence (AI), particularly machine learning and deep learning, has shown great promise in enhancing threat detection capabilities. This paper explores the integration of AI technologies into cybersecurity, reviewing current approaches including supervised, unsupervised, and semi-supervised learning, as well as deep learning methods such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). The paper evaluates various AI-based models for threat detection and compares their performance using standard datasets. Additionally, challenges in improving the real-time capability, interpretability, and generalization of AI models are discussed. Finally, the paper outlines future research directions, emphasizing the need for more diverse and larger datasets, and the potential of reinforcement learning and multimodal learning in building intelligent cybersecurity defense systems.

Keywords: Cybersecurity, Threat Detection, Artificial Intelligence.

1. Introduction

With the rapid advancement of information technology, cybersecurity threats have emerged as a significant challenge for modern society. While cyberattacks continue to evolve, traditional rule-based protection strategies can only manage certain known attack types and often struggle to counteract novel and complex threats. This limitation is particularly evident in detecting large-scale, high-frequency intrusions and advanced persistent threats (APTs). Consequently, developing effective methods to identify and mitigate new threats has become a critical focus in cybersecurity research. In recent years, artificial intelligence (AI) has opened up new opportunities in this domain. With notable achievements in various fields, machine learning and deep learning are proving increasingly valuable in enhancing threat detection and defense mechanisms.

Artificial intelligence, especially through machine learning and deep learning techniques, has found extensive application in network threat detection. By automating the processes of feature extraction and pattern recognition, AI systems can analyze existing data to identify potential attack patterns, significantly improving both detection accuracy and response speed. Commonly used methods in intrusion detection systems (IDS) include supervised and unsupervised learning within the realm of machine learning. Furthermore, advanced deep learning architectures, such as deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent neural networks (RNNs), harness their strong learning capabilities to achieve higher precision and robustness in threat detection.

However, despite the promising performance of AI in threat detection, several challenges remain. In particular, when handling massive changing network data, improving the generalization, real-time capability, and interpretability of AI models continues to be a stressful research issue.

Currently, many studies have begun to explore the applications of AI in cybersecurity. Abdullahi proposed an AI-based approach to IoT security, demonstrating AI's capability in detecting threats within complex environments [1]. Zeadally examined how advancements in AI can effectively enhance the defensive capabilities of cybersecurity systems, particularly against highly sophisticated

attacks [2]. Additionally, research by Weng and Wu indicated that deep learning-based techniques can significantly improve network data security, providing robust support against potential future cyberattacks [3].

However, despite the development of various effective techniques in existing studies, the application of AI in cybersecurity still faces numerous challenges. These include improving the generalization ability of models, optimizing training efficiency, and enhancing the real-time responsiveness of defense systems.

This paper seeks to explore the current advancements and developments in the application of artificial intelligence to cybersecurity threat detection. It emphasizes the utilization of diverse AI methodologies, such as machine learning, deep learning, reinforcement learning, and multimodal learning, highlighting their roles and contributions in this field. First, the paper will provide a detailed analysis of the application of different AI methods in threat detection, discussing their advantages, limitations, and current usage. Next, the paper will evaluate the detection performance of several models on standard datasets through comparisons, analyzing their applicability and limitations in different scales. Finally, it will examine how AI technologies can better drive the upgrading and optimization of cybersecurity defense systems.

2. Overview of Threat Detection Technologies

2.1. Machine Learning Methods in Threat Detection

2.1.1. Supervised Learning

Supervised learning, a machine learning approach, involves training a model on labeled datasets to enable it to predict labels for unseen data. In the field of cybersecurity, this method is frequently applied to tasks like malware identification and network intrusion detection. By examining data features paired with their corresponding labels, these models learn to classify new data into predefined categories. Commonly used algorithms in this context include Decision Trees, Support Vector Machines(SVM), and Random Forests.

In threat detection, supervised learning leverages its strong capabilities in classification and prediction to automatically learn from large amounts of data and identify potential security threats. For example, by analyzing both the static attributes and dynamic behaviors of malware, supervised learning models can classify new software samples, rate their threat levels, and determine whether they pose a security risk. Using supervised learning models, a comprehensive set of operational processes can be established, including five key modules: Identify, Protect, Detect, Respond, and Recover, to create a fully automated cybersecurity threat detection system [4].

While supervised learning demonstrates strong performance in threat detection, it relies heavily on a substantial amount of labeled data, which can be both expensive and time-intensive to acquire in the real world. Moreover, the scarcity of accessible and open-source threat intelligence databases poses an additional challenge.

2.1.2. Unsupervised Learning

Unlike traditional machine learning models, unsupervised learning requires little to no human intervention or "training" of the algorithm. It can automatically classify or cluster input data without the need for labeled data. The key distinction between supervised and unsupervised learning lies in the presence of supervision—i.e., whether the input data has labels. If the input data is labeled, it is considered supervised learning; if there are no labels, it falls under unsupervised learning.

Unsupervised learning relies on key algorithms such as clustering techniques, Principal Component Analysis(PCA), and Independent Component Analysis(ICA), among others. Clustering methods, in particular, are widely regarded as the most frequently utilized approach in unsupervised learning. These techniques aim to partition datasets into distinct clusters, revealing the hidden patterns and structures within the data.

Security personnel are often overwhelmed by a large volume of alerts, and some abnormal behaviors can be difficult to detect or are easily overlooked. Unsupervised learning methods can help make more efficient use of resources and improve detection accuracy. Moreover, unsupervised learning can serve as a proactive measure to prevent cyberattacks, enabling security teams to take a more active role in threat prevention and real-time response. In the field of cybersecurity, unsupervised learning is primarily applied to network behavior anomaly detection, identification of attack behaviors, user behavior analysis, and detection of malicious network code.

Unsupervised learning offers a valuable method for detecting cybersecurity threats, but it encounters several obstacles in real-world applications. A key challenge lies in the dependence on vast amounts of unlabeled data, where the quality of this data plays a critical role in determining the model's effectiveness. Ensuring access to high-quality unlabeled datasets is a persistent difficulty. Furthermore, unsupervised learning models are often perceived as "black boxes" due to their opaque decision-making processes, which lack transparency and interpretability. This limitation can result in challenges like false positives and false negatives when identifying anomalous activities.

2.1.3. Semi-supervised Learning

Semi-supervised learning integrates the benefits of supervised and unsupervised approaches by utilizing a limited amount of labeled data in conjunction with a substantial quantity of unlabeled data for training models. In the realm of cybersecurity, especially in threat detection, this method proves effective in mitigating the challenges posed by insufficient labeled data while exploiting unlabeled data to improve the model's generalization and predictive performance.

Key algorithms associated with semi-supervised learning include Autoencoders, Generative Adversarial Networks (GANs), and One-Class Support Vector Machines (OCSVM), among others. This approach provides a balanced solution, combining the strengths of supervised and unsupervised methodologies, though it is not without its drawbacks. The persistent shortage of labeled data continues to be a significant obstacle, particularly in cybersecurity, where accurately labeling malicious traffic remains a daunting task. Furthermore, akin to unsupervised methods, semi-supervised models—especially those based on deep learning—are often regarded as "black boxes," lacking clarity and interpretability in their decision-making processes, which can hinder trust and accountability.

2.2. Deep Learning Methods in Threat Detection

2.2.1. Deep Neural Networks (DNN)

DNNs are multi-layer feedforward neural networks that learn complex representations of data by stacking multiple hidden layers. In network threat detection, DNNs are capable of automatically extracting high-level features from network traffic to identify and classify malicious behaviors.

DNNs are trained using the backpropagation algorithm alongside gradient descent, where activation functions like ReLU introduce non-linearity, allowing the network to capture and model intricate patterns of cybersecurity threats. The typical workflow begins with preprocessing and extracting features from network traffic data. The DNN model, with its multi-layered architecture, is then trained to learn hierarchical representations of the data. Once trained, the model is deployed to analyze real-time traffic for detecting potential threats.

Despite their effectiveness, training DNNs comes with notable challenges. These models require vast amounts of labeled data and significant computational resources, which can be resource-intensive. Moreover, DNNs are often criticized for their lack of transparency, commonly referred to as the "black box" issue, as their internal decision-making processes are difficult to interpret and understand.

2.2.2. Convolutional Neural Networks (CNN)

CNNs are particularly effective in handling data with a grid-like topology, such as images and network traffic packets. A typical CNN architecture comprises an input layer, convolutional layers, pooling layers, and fully connected layers. Through the convolutional layers, CNNs automatically

extract local features, while pooling layers reduce the spatial dimensions of these features, enabling the model to capture patterns that are indicative of network attacks.

In the context of threat detection, CNNs first preprocess network traffic data, then leverage convolutional and pooling layers to identify and extract significant features. Subsequently, the fully connected layers perform the classification task, facilitating the identification and categorization of potential threats. This hierarchical approach allows CNNs to effectively discern complex patterns and anomalies within network traffic.

When CNNs are applied to network traffic data, they often face the challenge of class imbalance, where normal traffic significantly outnumbers anomalous traffic. This imbalance leads to poor recognition of the minority class (anomalous traffic), which severely impacts the model's training effectiveness and detection accuracy.

2.2.3. Recurrent Neural Networks (RNN)

RNNs and their advanced variants, including Long Short-Term Memory(LSTM) networks and Gated Recurrent Units(GRUs), are specifically tailored for processing sequential data, which makes them highly effective for detecting network attacks that unfold over time. RNNs leverage recurrent connections to capture temporal dependencies within the data, enabling the model to retain information from previous inputs and utilize this context to identify patterns in sequential or time-series-based attack behaviors. LSTM and GRU networks address the long-term dependency problem inherent in traditional RNNs by introducing gating mechanisms, which help the network retain and update important information over longer sequences. These features make RNN-based models highly effective for detecting complex, time-dependent patterns in network traffic.

In threat detection, RNNs first perform time-series analysis on network traffic data, then use the RNN structure to learn the temporal dynamics of the data, and finally make attack predictions based on the learned patterns.

RNNs can face challenges like vanishing or exploding gradients when handling long sequences of data, which can impede the model's capacity to effectively learn over extended time intervals. To address these challenges, careful design of the network structure and training strategies is required.

3. Literature Analysis and Comparison of Experimental Results

3.1. Experimental Results of Different Methods in Threat Detection

The following text primarily selects seven large models used for threat detection, which may vary in their application scope. The implementation techniques encompass machine learning, deep learning, and other technologies.

Al-Abassi A. et al. introduced an ensemble deep learning-based model for cyber-attack detection in Industrial Control Systems. This model integrates DNN with Decision Tree classifiers to identify network attacks from newly balanced datasets. The proposed approach achieved an accuracy of 0.96 and an F1 score of 0.9383 on the Gas Pipeline (GP) dataset, and an impressive accuracy of 0.9967 with an F1 score of 0.99 on the Secure Water Treatment (SWaT) dataset. The authors highlighted the advantages of their method, including improved F1 scores on imbalanced datasets and higher accuracy compared to traditional classifiers on both datasets [5].

Injadat M. et al. introduced an innovative multi-stage optimized machine learning-based framework for Network Intrusion Detection Systems(NIDS), which aims to minimize computational complexity without compromising detection performance. Their study examined the effect of oversampling techniques on the model's training sample size and determined the minimum effective sample size for training. The approach primarily incorporated three key techniques: SMOTE (Synthetic Minority Over-sampling Technique), feature selection, and hyperparameter optimization. The performance of the proposed multi-stage optimized machine learning framework was assessed using the CICIDS2017 and UNSW-NB15 datasets. The model achieved over 99% detection accuracy

on both datasets, and after hyperparameter optimization, the detection accuracy improved by 1-2% compared to recent works in the literature, with a 1-2% reduction in false positive rates [6].

Kim, A. et al. introduced an optimal Convolutional Neural Network and Long Short-Term Memory(CNN-LSTM) model, conducting a comparative analysis of the performance of CNN-LSTM, LSTM-CNN, and DNN models based on metrics such as accuracy, precision, and F-Score. The CNN-LSTM model demonstrated superior performance, especially in precision and F-Score, surpassing the other models. Furthermore, the model was validated using publicly available datasets, including CSIC-2010 and CICIDS2017, as well as real-time data. The model achieved high accuracy ranging from 91% to 93% on the CSIC-2010 and CICIDS2017 datasets, with precision between 86% and 98%, and an F-Score between 80% and 82% [7].

Su, T. et al. developed a traffic anomaly detection model, termed BAT-MC, which integrates Bidirectional Long Short-Term Memory(BLSTM) with an attention mechanism. The primary advantage of the BAT-MC model lies in its capacity to automatically extract features, thus eliminating the need for manual feature engineering, and enabling it to capture network traffic characteristics more comprehensively. The study evaluated the accuracy of the BAT-MC model against that of CNN and RNN models using the KDDTest+ and KDDTest-21 datasets. The BAT-MC model achieved an accuracy of 84.25% on the KDDTest+ dataset, outperforming the CNN model by 4.12% and the RNN model by 2.96%. On the KDDTest-21 dataset, the BAT-MC model surpassed the CNN model by 4.75% and the RNN model by 7.1% [8].

Abou El Houda, Z. et al. introduced a framework powered by Explainable Artificial Intelligence (XAI) for detecting IoT-related attacks, incorporating three XAI techniques—RuleFit, LIME, and SHAP. This framework exhibited exceptional performance on the NSL-KDD dataset, attaining an accuracy of 0.88, an F1-score of 0.88, precision of 0.96, and recall of 0.88. In comparison with conventional models, including SVM, CNNs, and other DL and ML approaches, this model outperformed in all key metrics. On the UNSW-NB15 dataset, the framework achieved a remarkable accuracy of 0.99 and a True Positive Rate(TPR) of 0.99, surpassing the performance of other selected ML/DL-based models [9].

Liu, L. et al. investigated the application of machine learning and deep learning techniques for intrusion detection in imbalanced network traffic, introducing a novel Difficulty Set Sampling Technique(DSSTE) algorithm to tackle the issue of class imbalance. Through extensive experimentation, Liu demonstrated that the DSSTE algorithm outperformed 24 alternative methods in terms of overall performance. The algorithm exhibited robust results on both the NSL-KDD and CSE-CIC-IDS2018 datasets, achieving accuracy and F1-scores exceeding 96% on each dataset [10].

Kim, J. et al. proposed an intrusion detection model leveraging deep learning techniques, with a specific emphasis on detecting denial of service(DoS) attacks. The researchers designed 18 distinct experimental setups, varying key hyperparameters such as image formats(RGB vs. grayscale), the number of convolutional layers, and kernel dimensions. The results of these experiments highlighted the model's performance across a range of configurations, focusing on metrics like accuracy and F1-score. For the KDD dataset, the CNN model achieved accuracy rates of 99% or higher in both binary and multi-class classification tasks, whereas the RNN model attained 99% accuracy in binary classification and 93% in multi-class classification. On the CSE-CIC-IDS 2018 dataset, the CNN model demonstrated an average accuracy of 91.5%, while the RNN model exhibited a lower accuracy, averaging 65% [11].

The following summarizes the results obtained from testing the selected models on different datasets. It is important to note that in these tables, any model or dataset that was not used has been omitted by the author.

The performance of each method and model on various datasets, including metrics such as accuracy, precision, recall, and F1-score, is presented in Tables 1-4:

Table 1. Accuracy of Different Methods on Different Datasets

	KDD	NSL-KDD	KDDTest+	KDDTest-21	CSE-CIC-IDS2018	CICIDS 2017	CSIC-2010	UNSW-NB15	GP	SWaT
a DL model in ICS									0.96	0.9967
multi-stage optimized ML-based NIDS framework						>0.99		>0.99		
CNN-LSTM						0.93	0.9154			
BAT-MC			0.8425	0.6942						
XAI-enpowered framework for IoT attacks			0.88					0.99		
DSSTE		0.8069			0.9629					
DL (CNN)-based intrusion detection against DoS attacks	>0.99				≈0.915					

Table 2. Precision of Different Methods on Different Datasets

	KDD	NSL-KDD	KDDTest+	CSE-CIC-IDS2018	CICIDS 2017	CSIC-2010	UNSW-NB15	GP	SWaT
a DL model in ICS								0.9463	0.97
multi-stage optimized ML-based NIDS framework					>0.98		0.99		
CNN-LSTM					0.8647	0.9854			
XAI-enpowered framework for IoT attacks			0.96						
DSSTE		0.8069		0.9629					
DL (RNN)-based intrusion detection against DoS attacks	>0.99 (binary-class) >0.77 (multi-class)								

Table 3. Recall of Different Methods on Different Datasets

	KDD	KDDTest+	CICIDS2017	CSIC-2010	UNSW-NB15	GP	SWaT
a DL model in ICS						0.9372	0.99
multi-stage optimized ML-based NIDS framework			0.99		0.99		
CNN-LSTM			0.7683	0.6826			
XAI-enpowered framework for IoT attacks		0.88					
DL (RNN)-based intrusion detection against DoS attacks	1 (binary-class) >0.71 (multi-class)						

Table 4. F-score (F1-score) of Different Methods on Different Datasets

	KDD	NSL-KDD	KDDTest+	CSE-CIC-IDS2018	CICIDS 2017	CSIC-2010	UNSW-NB15	GP	SWaT
a DL model in ICS								0.9383	0.99
multi-stage optimized ML-based NIDS framework					0.99		0.99		
CNN-LSTM					0.8136	0.8056			
XAI-enpowered framework for IoT attacks			0.88						
DSSTE		0.7934		0.9629					
DL(CNN)-based intrusion detection against DoS attacks				≈0.915					
DL (RNN)-based intrusion detection against DoS attacks	>0.99 (binary-class) >0.80 (multi-class)								

Some researchers did not test certain metrics, such as Recall, for some models, so these have been omitted from the table. Blank spaces indicate that the corresponding value is not available.

3.2. The impact of dataset differences on experimental results

For instance, in the aforementioned XAI framework, the NSL-KDD and UNSW-NB15 datasets exhibit distinct variations in feature distribution and types of attacks. The NSL-KDD dataset not only resolves the redundancy issue inherent in the KDD Cup 1999 dataset but also ensures a balanced number of records in both the training and testing sets. Conversely, the UNSW-NB15 dataset encompasses a wider range of attack types and a larger volume of data, which may influence the model's performance across different datasets.

Regarding the CNN-LSTM model, the experimental findings suggest that the model's efficacy is closely linked to the sample size and the diversity of the training data. Given that the CSIC-2010 and CICIDS2017 datasets contain relatively few samples, this may result in overfitting, leading to diminished performance. Access to a larger volume of non-redundant HTTP data could potentially enhance the model's performance, yielding more robust results.

Referring to the multi-stage optimized machine learning-based NIDS framework, the CICIDS 2017 and UNSW-NB 2015 datasets employed in the framework vary in terms of attack types and the non-linearity of their features, which in turn impact both the model's performance and the choice of techniques. The experimental outcomes reveal that the SMOTE technique helps reduce the training sample size, feature selection minimizes the feature set, and hyperparameter optimization improves the model's detection capabilities. Despite the disparities in feature distribution between the two datasets, the proposed framework demonstrates strong performance on both.

In summary, the differences in feature distribution and attack types between different datasets (such as NSL-KDD and UNSW-NB15) can lead to inconsistent model performance across datasets. Datasets with fewer samples (such as CSIC-2010 and CICIDS2017) may cause the model to overfit, reducing its generalization ability and performance. The differences in feature distribution and attack types across datasets require specific optimizations for the model. However, despite these differences in feature distribution, through appropriate technical adjustments, the model can perform well on different datasets.

4. Challenges and Outlooks

4.1. Challenges

4.1.1. The gap between model performance and real-world requirements

While certain deep learning models, such as DSSTE, achieve exceptional performance on benchmark datasets with metrics like accuracy and F1-score, they remain largely untested in real-world network environments. Novel attack techniques may lead to outcomes that fail to meet anticipated levels of effectiveness. Moreover, the dynamic and complex nature of real-world data often impacts the consistency of these models, posing challenges to maintaining their high performance outside controlled conditions. Consequently, further evaluation and validation in practical environments are essential for their deployment in real-world scenarios.

4.1.2. Limitations in datasets

The public available datasets used for threat detection, such as NSL-KDD and CSIC-2010, still have limitations in terms of attack type coverage, sample size, and distribution balance. For instance, the CSIC-2010 and CICIDS 2017 datasets have relatively small sample sizes, which can lead to overfitting and reduced generalization capability. Moreover, some datasets have a limited number of attack types or an imbalanced distribution, which may restrict the model's applicability in more complex environments. The CSIC-2010 dataset, for example, has a small sample size and covers only a limited range of attack types, mainly focusing on HTTP attacks at the web application layer (such as SQL injection and directory traversal). This restricts the model's applicability when dealing with attacks from other network layers, like DDoS or network scanning. While the NSL-KDD dataset addresses the redundant records issue of the KDD 99 dataset, it still only includes a narrow set of attack types, covering basic attacks such as probing, DoS, U2R, and R2L. These attack types, however, do not adequately represent modern, complex network threats. For example, advanced persistent threats (APTs) or sophisticated IoT-based attacks are not included.

In contrast, the CSE-CIC-IDS2018 dataset performs better in this regard, as it not only includes most of the basic attack types but also covers modern attack types such as DDoS, brute force, malware, and identity spoofing. Moreover, its data distribution is more representative of real-world network environments. However, despite its large sample size, the dataset still faces the issue of data imbalance, which could affect training performance and should be carefully considered.

4.2. Future outlooks

Future research should focus on creating more extensive and diverse testing datasets that encompass a broader spectrum of attack types, thereby improving the generalization capabilities of models in complex environments. Additionally, fostering the creation of data-sharing platforms would facilitate global collaboration among researchers, accelerating advancements in cybersecurity model development. Moreover, enhancing the real-time responsiveness of models to dynamic network data is a critical avenue for future progress. This could be achieved by integrating emerging technologies, such as reinforcement learning and multimodal learning, to bolster model performance in complex and ever-evolving contexts.

5. Conclusion

This paper reviews the current application and development direction of artificial intelligence technologies in network security threat detection. By exploring core methods such as supervised learning, unsupervised learning, deep learning, and multimodal learning, and combining recent experimental studies and model performance evaluations, this paper highlights the technical challenges in the field and forecasts future trends.

The experimental results emphasize the importance of dataset diversity and quality in influencing model performance. The scale, coverage of attack types, and sample distribution balance of existing

public datasets still need further improvement to better adapt to the complex threats in modern network environments. At the same time, the robustness and generalizability testing of models in real-world environments require strengthening.

Future research should focus on the following directions: First, developing more comprehensive and large-scale datasets that are closer to real-world environments to enhance the generalization capability of models in complex network threat scenarios; second, investigating the application of cutting-edge technologies like reinforcement learning and multimodal learning in threat detection aims to establish a more robust and intelligent network security defense framework.

References

[1] Abdullahi M, Baashar Y, Alhussian H, Alwadain A, Aziz N, Capretz L F, & Abdulkadir S J. Detecting cybersecurity attacks in internet of things using artificial intelligence methods: A systematic literature review. Electronics, 2022, 11 (2), 198.

[2] Zeadally S, Adi E, Baig Z, & Khan I A. Harnessing artificial intelligence capabilities to improve cybersecurity. Ieee Access, 2020, 8, 23817 - 23837.

[3] Weng Y, & Wu J. Leveraging artificial intelligence to enhance data security and combat cyber-attacks. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006 - 4023, 2024, 5 (1), 392 - 399.

[4] Manoharan A, & Sarker M. Revolutionizing Cybersecurity: Unleashing the Power of Artificial Intelligence and Machine Learning for Next-Generation Threat Detection. DOI: https://www.doi.org/10.56726/IRJMETS32644, 1, 2023.

[5] Al-Abassi A, Karimi pour H, Dehghan Anha A, & Parizi R M. An ensemble deep learning-based cyber-attack detection in industrial control system. Ieee Access, 2020, 8, 83965-83973.

[6] Injadat M, Moubayed A, Nassif A B, & Shami A. Multi-stage optimized machine learning framework for network intrusion detection. IEEE Transactions on Network and Service Management, 2020, 18 (2), 1803 - 1816.

[7] Kim A, Park M, & Lee D H. AI-IDS: Application of deep learning to real-time Web intrusion detection. IEEE Access, 2020, 8, 70245 - 70261.

[8] Su T, Sun H, Zhu J, Wang S, & Li Y. BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset. IEEE Access, 2020, 8, 29575 - 29585.

[9] Abou El Houda Z, Brik B, & Khoukhi L. “why should i trust your ids?”: An explainable deep learning framework for intrusion detection systems in internet of things networks. IEEE Open Journal of the Communications Society, 2022, 3, 1164 - 1176.

[10] Liu L, Wang P, Lin J, & Liu L. Intrusion detection of imbalanced network traffic based on machine learning and deep learning. IEEE access, 2020, 9, 7550 - 7563.

[11] Kim J, Kim J, Kim H, Shim M, & Choi E. CNN-based network intrusion detection against denial-of-service attacks. Electronics, 2020, 9 (6), 916.