Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 31 No. 8s (2024)

Network Security by Modified Teacher Learning Optimized Features and Ensemble Model

[1]Apoorva Patil, [2.] Dr. Ankit Pandit, [3.]Dr. Sanjeev Kumar Gupta, [4.] Dr. Laxmi Singh

Research scholar Associate Professor Dean Faculty of Engineering HoD ECE

[1]RNTU Bhopal (M.P) RNTU,

[2]Bhopal (M.P) RNTU,

[3]Bhopal (M.P) RNTU,

[4]Bhopal (M.P)

apoorva4589@gmail.com, ankit.pandit@aisectuniversity.ac.in, sanjeevkumar.gupta@aisectuniversity.ac.in laxmi.singh@aisectuniversity.ac.in

Article History:

Received: 12-10-2024

Revised: 15-11-2024

Accepted: 01-12-2024

Abstract:

Network nodes have limited energy hence attacks play an role that directly decreases the life span of network. It was found that these kind of network is vulnerable and may get attacked easily hence researcher developed alarming security models. Most of models are lacking in achieving the accuracy of alarming the attack class. This paper has proposed a model that filter input training data by modified teacher learning model. Further filtered features were transformed into relevant pattern by use of Nonnegative Matrix Factorization. Transformed network observation values were used for the training of ensemble model. Experiment was done on real network dataset and result shows that use of modified teacher learning and Nonnegative matrix factorization has increases the alarming accuracy of the attack class as well.

Keywords: Deep Learning, Intrusion Detection, Feature Optimization, Genetic Algorithm, Soft Computing.

I. Introduction

As communication technology continues to evolve, network frameworks have become essential for the seamless transportation of diverse and heterogeneous information across distributed environments [1]. Among these, Wireless Sensor Networks (WSNs) play a critical role in enabling communication through interconnected sensor nodes, which can be deployed in various topologies such as star, tree, or mesh configurations [2]. These networks are designed to sense, collect, and transmit data across wireless channels while efficiently managing network flow. The primary functional capabilities of WSNs include sensing, processing, computation, and communication, making them highly versatile for a range of applications [3].

WSNs are widely utilized for monitoring, security surveillance, and tracking, offering a cost-effective platform that operates with limited energy resources while ensuring effective wireless data transmission [4]. However, despite their advantages, WSNs remain vulnerable to unauthorized access and cyber threats, which can originate from both internal and external sources. Intruders may exploit

security weaknesses in the network, necessitating the implementation of robust protection mechanisms for securing sensor nodes and ensuring network integrity.

With the growing attack surface in modern networks, concerns over cybersecurity have escalated, leading to extensive research on network intrusion detection systems (NIDSs) aimed at mitigating these threats [5]. Among the different approaches to intrusion detection, AI-driven anomaly detection frameworks have gained significant attention in recent years due to their ability to identify novel and sophisticated attacks. Several techniques have been proposed to enhance the performance of NIDS, improving their detection accuracy and adaptability. However, one of the major challenges that persist in AI-based NIDS is data imbalance, which adversely affects the ability of machine learning models to effectively recognize malicious behaviors. This issue arises because network datasets predominantly consist of legitimate traffic, while malicious activities leading to service disruptions are relatively rare [6]. Furthermore, the dataset may include well-documented attack signatures, whereas zero-day attacks or previously unseen attack patterns are significantly underrepresented, making it difficult for AI models to generalize effectively [7].

An IDS (IDS) is a security mechanism designed to monitor network traffic and detect malicious activities by identifying deviations from expected behavior. IDS can be categorized into signature-based and anomaly-based detection. The signature-based approach relies on predefined attack patterns, offering high accuracy and low false positive rates for known threats but failing to detect zero-day attacks without frequent updates. In contrast, the anomaly-based approach uses statistical methods and machine learning algorithms to identify unusual network behavior, allowing it to detect unknown threats. However, it often suffers from higher false positive rates due to the dynamic nature of legitimate traffic patterns [8].

The evolution of machine learning (ML) and deep learning (DL) techniques has significantly enhanced IDS capabilities, enabling systems to adapt and learn autonomously from network behavior. Deep learning models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have proven highly effective in detecting both known and emerging threats by recognizing complex traffic patterns. CNNs excel in extracting spatial features, while RNNs capture temporal dependencies, improving detection accuracy. These advancements have made AI-driven IDS solutions more efficient and robust, addressing the growing challenges of modern cybersecurity threats [9].

This paper has identify few issues:

- Most of work only identify attack and non-attack class of network session.
- Further calculation of most of work is complex and time taken.
- Improves the true alarm rate with intrusion class efficiency.

II. Related Work

Papamartzivanos et al. [10] introduced an advanced approach utilizing the MAPE-K framework to develop a misuse-based IDS with scalability, self-adaptability, and autonomy. Their method is capable of extracting generalized features for problem reconstruction, even when faced with unknown environments and unlabeled data. The researchers argue that their approach is effective in identifying

variant attacks by understanding their inherent nature. They further conducted experiments demonstrating that their method can handle novel attack scenarios without the need for manual updates to the training dataset.

Qaddoura et al. [11] proposed a deep multilayer intrusion detection method incorporating two recognition stages—first, detecting an intrusion's presence, and second, identifying the specific type of attack. Additionally, the study introduced an oversampling technique to ensure improved detection accuracy. Extensive experiments were conducted across multiple conditions for both recognition phases, along with an evaluation of two different oversampling strategies to enhance the quality of classification results.

Yao et al. [12] explored a technique combining Principal Component Analysis (PCA) with a Deep Convolutional Neural Network (DCNN) for detecting Denial-of-Service (DoS) attacks in Wireless Sensor Networks (WSNs), which often suffer from storage constraints. Their approach demonstrated superior feature extraction capabilities compared to other deep learning architectures, achieving higher detection accuracy. Their findings emphasized that integrating learning-based techniques is an effective strategy for detecting and preventing DoS attacks in WSNs while considering the resource limitations of these devices.

Gebremariam et al. [13] introduced a neural network-based IDS using a Multi-Layer Perceptron (MLP) Artificial Neural Network (ANN). The model was trained on labeled network traffic data to recognize attack patterns and later applied to real-time traffic for attack detection and localization. The evaluation, conducted through simulated attack scenarios, confirmed the model's effectiveness in both identifying and pinpointing attacks in WSNs. Their findings highlighted the potential of ANNs in enhancing network security by improving attack detection and localization.

Chen et al. [14] developed HTTPSmell, a novel method for detecting malicious HTTP traffic by leveraging a dataset of XSS attacks sourced from GitHub. Their framework automatically applies a deep learning model with high generalization ability, even when trained on limited data. Their approach incorporates semi-supervised learning, utilizing Unsupervised Data Augmentation (UDA) and Keyword Library Avoidance (KLA)-based augmentation, leading to higher generalization, broader scenario coverage, and improved detection efficiency, particularly in small datasets.

Dash et al. [15] proposed an optimized Long Short-Term Memory (LSTM) model for network anomaly detection, employing three optimization techniques—Particle Swarm Optimization (PSO), JAYA, and the Salp Swarm Algorithm (SSA)—to fine-tune the LSTM hyperparameters. Their research utilized datasets such as NSL-KDD, CICIDS, and BoT-IoT, demonstrating that optimization enhances anomaly detection accuracy.

Ramu et al. [16] introduced a meta-heuristic optimization and deep learning-based method to enhance network IDSs (NIDS). Their framework begins with data pre-processing for standardization and balancing, followed by feature selection using an Extended Pelican Optimization Algorithm (Ex-Pel). Finally, the Self-Attention Assisted Weighted Autoencoder (SAttn_WAE) is applied to detect attacks with higher precision by focusing on optimal feature selection.

III. Proposed Model

This section provides a comprehensive overview of the Modify Teacher Optimization Feature Network Security (MTOFNS) model. Figure 1 illustrates its core stages, including dataset preprocessing, dimensionality reduction, feature transformation, and training. The preprocessing phase ensures data consistency, while dimensionality reduction removes redundant features to improve efficiency [12, 14]. Feature transformation refines data attributes for better classification, and the training stage employs machine learning techniques for accurate intrusion detection. Each of these components is discussed in detail under separate headings in this section, explaining their significance in optimizing MTOFNS for enhanced network security and performance.

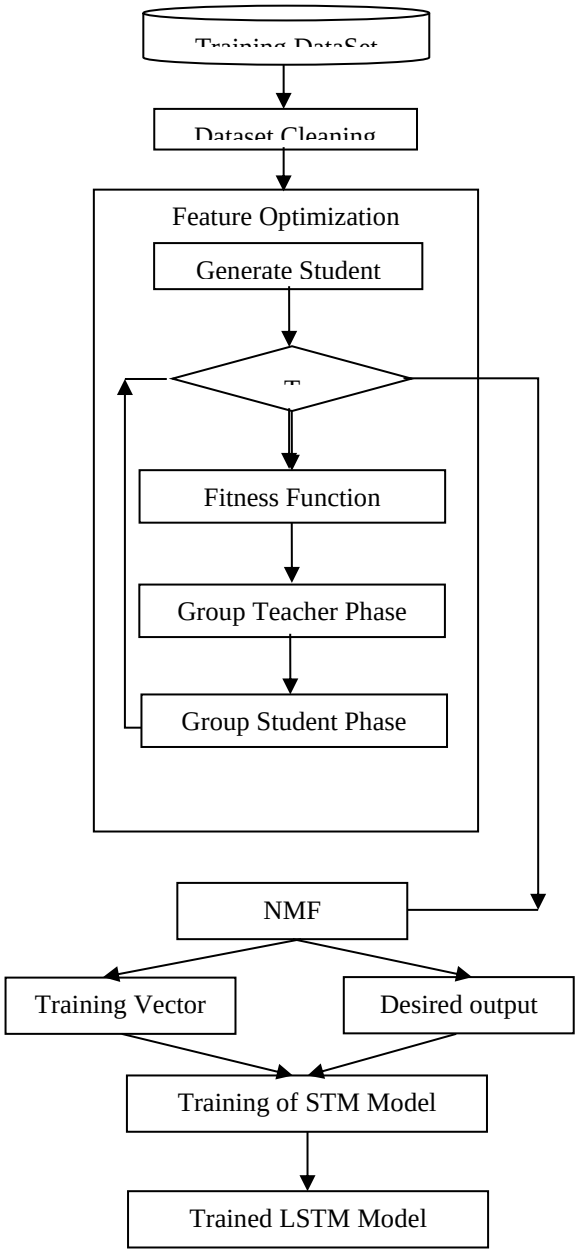

Fig. 1 Block diagram of MTOFNS network intrusion detection.

Dataset Processing

The input dataset used for malicious session detection contains numerous features, each with varying levels of significance. In this initial stage, the dataset undergoes a cleaning process to remove irrelevant or redundant information, ensuring that only meaningful data is used for further analysis. For instance, the dataset employed in this research contains several fields, but the first few feature values were deemed unnecessary for detecting malicious sessions [15]. Examples of such fields include session IDs, connection types, and transferring protocols, which do not contribute significantly to identifying malicious activity. By eliminating these features, the dataset becomes more focused and efficient, allowing the model to concentrate on the most critical features.

PNSD←DataProcessing(NSD) ----------Eq. 1

In Equation 1, the Network Session Dataset (NSD) represents the raw input data, while the Processed NSD (PNSD) is the refined version after initial preprocessing. The processed dataset is structured in a matrix format, where each row corresponds to a network session, and each column represents a specific feature associated with that session.

Feature Optimization Following the data cleaning phase, the PNSD matrix undergoes feature optimization, leveraging the Modified Teacher Learning Optimization (MTLO) algorithm [18]. This step aims to reduce the number of features in the training vector while improving the model's accuracy. By selecting the most relevant features, MTLO enhances the overall efficiency and effectiveness of intrusion detection.

Modified Teacher Learning Optimization Algorithm (MTLO)

The MTLO algorithm is inspired by the natural teaching-learning process. In this approach, each chromosome represents a student, who strives to achieve higher scores based on the knowledge imparted by the teacher. Traditionally, the best local solution acts as the teacher, guiding students toward an optimal solution. However, in this modified version, students are instructed by multiple teachers, forming a group-based learning approach. This modification allows students to learn from a diverse set of optimal solutions, leading to a more refined feature selection process.

Generating Student Population The student population consists of multiple students (chromosomes), where each chromosome represents an optimized feature set. Each student is encoded as a vector with 'n' elements, where 'n' corresponds to the number of columns in the PNSD matrix. The elements in this vector are binary (0 or 1):

- 1 indicates that a feature is included for training.
- 0 means the feature is excluded.

To create a diverse range of student populations, 's' students are generated, forming a student population matrix (SP) of size $s \times n$. Feature selection is performed using a Gaussian random value generator, ensuring a randomized yet effective selection of features for optimization.

SP←GeneratePopulation(s, n, f)----------Eq. 2

Table 1 Notation of MTOFNS.

Notations	**Meaning**
NSD	Network Session Dataset
PNSD	Processed Network Session Dataset
MTLO	Modified Teacher Learning Optimization
SP	student population
s	Number of Students
n	Number of Features
f	Number of minimum features present in student vector
SF	Student Fitness
DOC	Desired Output Class
SPb	Student Phase best
FSD	Filter Session Dataset
W	NFM base Matrix
H	NFM output Coefficient Matrix
r	Reduce Dimension
TEM	Trained Ensemble Model

Fitness Function Each student (chromosome) is assessed based on its fitness, which is determined by evaluating how well its selected feature set enhances detection accuracy. This is accomplished through a fitness function that measures the "distance" between the student's feature selection and an optimal solution. The student's feature vector is fed into a learning model network, which is trained on the processed dataset (PNSD) to assess its effectiveness in identifying malicious sessions. The detection accuracy obtained from this evaluation serves as the fitness score (SF) for each student.

SF←learningAccuracy(SP,PNSD, DOC)

Update Student position After calculating each student's fitness value, the population is sorted in descending order based on fitness scores. The best-performing student is identified as the teacher, representing the most optimized feature set. To further refine the population, a genetic-inspired optimization strategy is applied through the Teacher and Student phases, allowing for continuous improvement in feature selection.

Modify Teacher Phase In this step, students are taught by a group of teachers rather than a single best teacher. Each teacher is randomly assigned a group of students, increasing the probability of finding an optimized feature set in less time. A specified number of groups (G) is formed. In each group, G random positions in the student feature vectors are flipped (changing 0 to 1 or 1 to 0), except for the

best-performing student, which remains unchanged. This ensures that the algorithm continues to explore new feature combinations while preserving the best solution. Once modifications are made, students are re-evaluated for fitness. If a modified student performs better than its parent, it replaces the parent in the population. If no improvement is observed, the parent remains unchanged.

SP←Teacher_Phase(SP, SF, PNSD, DOC)

Student Phase In the Student Phase, students are randomly grouped, and the best-performing student within each group is designated as the teacher. The process mirrors the Modify Teacher Phase, where random positions in student vectors are flipped, except for the best local student. This iterative refinement continues until the maximum number of iterations is reached, ensuring that the feature selection process evolves toward optimal accuracy.

SP←Student_Phase(SP, SF, PNSD, DOC)

Feature Selection Once the iterative process is complete, the best student is chosen from the final population. The features with a value of 1 in the student vector are selected as the most relevant features for the training vector, while features with a value of 0 are excluded. Additionally, the desired output matrix is updated at this stage to reflect the final set of selected features.

This process ensures that the feature set is optimized for training, improving both the efficiency of the detection system and the accuracy of malicious session identification in cloud environments.

FSD←Filter(PSD,SPb)

Nonnegative Matrix Factorization (NMF) Feature optimization plays a crucial role in machine learning and data analysis, where dimensionality reduction techniques help improve model performance and interpretability [20, 21]. Nonnegative Matrix Factorization (NMF) is a powerful feature extraction method that decomposes a nonnegative data matrix into lower-dimensional representations while preserving its inherent structure.

NMF is particularly useful in applications where data components are naturally nonnegative, such as image processing, text mining, and bioinformatics. By factorizing an input matrix FSD into two nonnegative matrices W and H, such that:

[W H]=Non_Negative_Matrix_Factorization(FSD, r)

W is the basis matrix (m×r),

H is the coefficient matrix (r×n),

r is the reduced dimensionality.

NMF ensures that the resulting factors are interpretable, meaning that each feature is expressed as an additive combination of nonnegative basis components [20]. This makes NMF highly effective for feature selection and optimization, as it eliminates redundancy and enhances meaningful patterns in the data.

Ensemble Model Learning model is a combination of two different set where input features were used for the training of models separately and gives an output of class collectively. This work uses LSTM [22] and Neural Network for the learning takes H, DOC as input. LSTM have four layers for learning.

Input Layer: This layer takes the preprocessed network data as input. The input features can include a range of attributes such as packet size, time intervals, protocol type, source and destination IP addresses, etc.

Embedding Layer (Optional): If the input data includes categorical features, an embedding layer can be used to transform these features into continuous vectors that are easier for the LSTM to process [23].

LSTM Layers: One or more LSTM layers are used to learn the temporal patterns in the data. The number of LSTM layers and the number of neurons in each layer can be adjusted based on the complexity of the data and the desired level of model expressiveness.

Dropout Layer: To prevent over fitting, dropout layers are often added after the LSTM layers. Dropout layers randomly deactivate a fraction of neurons during training, which helps in generalizing the model better to unseen data.

Proposed **MTOFNS** Algorithm

Input: NSD

Output: TEM // Trained Ensemble Model

1. [PNSD DOC]←DataProcessing(NSD) // DO: Desired output class
2. SP←GeneratePopulation(s, n, f)
3. Loop 1:itreration
4. SF←LSTMAccuracy(SP,PNSD, DOC)
5. **SP←Teacher_Phase(SP, SF, PNSD, DOC)**
6. **SP←Student_Phase(SP, SF, PNSD, DOC)**
7. EndLoop
8. SF←LSTMAccuracy(SP,PNSD, DOC)
9. **SPb** ←Best(SF)
10. **FSD←Filter(PSD,SPb)**
11. H=Non_Negative_Matrix(FSD, r)
12. MLSTM←Intialize_LSTM(TF, PNSD)
13. MNN←Intialize_LSTM(TF, PNSD)
14. TEM←Train_Ensemble(MLSTM,MNN, H, DOC)

Above algorithm has taken an input of the raw network sessions and find the features that are effective by use of modified teacher learning optimization and transformed features by nonnegative matrix factorization method for the training of ensemble model.

IV. EXPERIMENT AND RESULTS

The proposed **MTOFNS** model and the comparative model were implemented using MATLAB software. The experiments were conducted on a system equipped with a 6th generation Intel Core i3 processor and 4 GB of RAM. The dataset used for evaluation was sourced from [23]. To assess the performance of **MTOFNS**, its results were compared with those of the cloud malicious session detection model proposed in [12].

Evaluation Parameter

Precision = TP / (TP+ FP)

Recall = TP / (TP + TN)

F-measure = 2 * Precision * Recall / (Precision + Recall)

Accuracy = (TP+TN)/(TP+TN+FP+FN)

Where

TP : True Positive, TN : True Negative, FP: False Positive, FN: False Negative

Execution Time: Algorithm success depends on retrieval time of data after user query processing. So this parameter is in seconds lessen value is desirable.

Results

Table 2 Network security alarm models comparison on the basis of precision values.

Testing Dataset	MTOFNS	DCNN
5000	0.9671	0.8254
10000	0.969	0.8082
15000	0.9689	0.7605
20000	0.9689	0.7584
25000	0.9686	0.7546

Table 1 shows precision values of network security alarming models. Proposed MTOFNS model has improved the true alarm class detection precision values by 19.31% as compared to DCNN [12]. It was found that use of modified teacher learning group phase model has increases the feature selection quality and results in better detection of attacks.

Table 3 Network security alarm models comparison on the basis of recall values.

Testing Dataset	MTOFNS	Previous Model
5000	0.9653	0.8467
10000	0.9672	0.7771
15000	0.9673	0.7892
20000	0.9649	0.7564
25000	0.964	0.7589

Table 3 shows recall values of network security alarming model for different testing datasets. It was found that use of Non-Negative Factor Matrix for feature transformation has increases the detection

class of ensemble model. It was found that MTOFNS has increases the recall value 18.64% as compared to DCNN [12].

Table 4 Network security alarm models comparison on the basis of f-measure values.

Testing Dataset	MTOFNS	Previous Model
5000	0.9662	0.8359
10000	0.9681	0.7923
15000	0.9681	0.7746
20000	0.9669	0.7574
25000	0.9663	0.7568

F-measure values of network security alarm models shown in table 4 for different dataset size. It was found that use of teacher learning and non-negative factor matrix has improved the input features learning matrix. Further it was found that use of feature optimization model has increases the f-measure values by 18.99%. Detection of attack clas if almost same in all set of experimental datasets.

Table 5 Network security alarm models comparison on the basis of accuracy values.

Testing Dataset	MTOFNS	Previous Model
5000	96.7145	82.5369
10000	96.9016	80.824
15000	96.8856	76.0467
20000	96.8861	75.8373
25000	96.8610	75.4605

Session attack detection accuracy values were shown in table 5. It was found that proposed MTOFNS has improved the detection accuracy of each experimental dataset. Further it was found that use of genetic model for feature selection with non-negative factor has successfully increased the work ensemble model learning. Combination of LSTM model with neural network in proposed work has increases the detection accuracy value by 19.317% as compared to DCNN model proposed in [12].

Table 6 Network security alarm models comparison on the basis of DOS attacks detection accuracy values.

Testing Dataset	MTOFNS	Previous Model
5000	96.1580	95.1508
10000	96.3249	96.1681
15000	96.9274	96.5217
20000	95.916	94.8133
25000	95.9061	94.7383

Table 7 Network security alarm models comparison on the basis of Probe attacks detection accuracy values.

Testing Dataset	MTOFNS	Previous Model
5000	85.0325	82.0225
10000	85.226	87.3451
15000	85.0835	85.0758
20000	84.9246	82.2944
25000	84.5714	83.7887

Table 8 Network security alarm models comparison on the basis of U2R attacks detection accuracy values.

Testing Dataset	MTOFNS	Previous Model
5000	74.4186	58.7226
10000	74.0741	86.8732
15000	70.2479	65.9454
20000	72.327	65.3231
25000	68.1159	62.6749

Table 6, 7 and 8 shows different attack class detection accuracy values for all set of experiment. Result shows that attack class detection of DOS is highly accurate but probe and U2R is less. Model has shown that proposed work has increases the work attack detection average accuracy value by 2.19% as compared to DCNN model.

IV. Conclusion

Network nodes operate with limited energy, making them vulnerable to attacks that significantly reduce their lifespan. These networks are easily compromised, prompting researchers to develop various security models. However, many existing models fail to accurately classify different types of attacks, often identifying only whether an attack occurred, without specifying its nature. This paper proposes an improved intrusion detection model that enhances input data using a Modified Teacher Learning approach. The selected features are further processed through Nonnegative Matrix Factorization (NMF) to extract relevant patterns. These transformed patterns are then used to train an ensemble model for effective classification. The proposed model addresses several issues: the binary classification limitation of existing methods, the complexity and time consumption of calculations, and the need to improve true alarm rates for specific intrusion classes. The MTOFNS model includes key stages such as preprocessing, dimensionality reduction, feature transformation, and machine learning-based training to optimize security and performance.

References

[1] Sections Figures References Ashiku, L., Dagli, C.: Network intrusion detection system using deep learning. Proc. Comput. Sci. 185, 239–247 (2021)

[2] K. Haseeb, N. Islam, T. Saba, A. Rehman, and Z. Mehmood, ‘‘LSDAR: A light-weight structure based data aggregation routing protocol with secure Internet of Things integrated next-generation sensor networks,’’ Sustain. Cities Soc., vol. 54, Mar. 2020, Art. no. 101995.

[3] R. Guetari, H. Ayari, and H. Sakly, ‘‘Computer-aided diagnosis systems: A comparative study of classical machine learning versus deep learningbased approaches,’’ Knowl. Inf. Syst., vol. 65, no. 10, pp. 3881–3921, Oct. 2023.

[4] R. Ramadan and K. Medhat, ‘‘Intrusion detection based learning in wireless sensor networks,’’ PLOMS AI, vol. 2, no. 1, pp. 1–20, 2022.

[5] Siddiqi, M.A., Pak, W.: Optimizing filter-based feature selection method flow for intrusion detection system. Electronics 9(12), 2114 (2020)

[6] Shyla, S., Bhatnagar, V., Bali, V., Bali, S.: Optimization of intrusion detection systems determined by ameliorated HNADAM-SGD algorithm. Electronics 11(4), 507 (2022)

[7] Toldinas, J., Venckauskas, A., Damaevicius, R., Grigaliunas, N.M., Baranauskas, E.: A novel approach for network intrusion detection using multistage deep learning image recognition. Electronics 10(15), 1854 (2021)

[8] Md Mahbubur Rahman, Shaharia Al Shakil, Mizanur Rahman Mustakim. "A survey on intrusion detection system in IoT networks, Cyber Security and Applications", Volume 3, 2025.

[9] Boiko, A. & Shendryk, V. System integration and security of information systems. Procedia Computer Science 104, 35–42 (2017).

[10] Papamartzivanos D., Gomez Marmol F., and Kambourakis G., Introducing Deep Learning Self-Adaptive Misuse Network Intrusion Detection Systems, 2019, IEEE Access, Piscataway, NJ, USA.

[11] R. Qaddoura, A. M. Al-Zoubi, H. Faris, and I. Almomani, ‘‘A multilayer classification approach for intrusion detection in IoT networks based on deep learning,’’ Sensors, vol. 21, no. 9, p. 2987, Apr. 2021.

[12] C. Yao, Y. Yang, K. Yin, and J. Yang, ‘‘Traffic anomaly detection in wireless sensor networks based on principal component analysis and deep convolution neural network,’’ IEEE Access, vol. 10, pp. 103136–103149, 2022.

[13] G. G. Gebremariam, J. Panda, and S. Indu, ‘‘Localization and detection of multiple attacks in wireless sensor networks using artificial neural network,’’ Wireless Commun. Mobile Comput., vol. 2023, pp. 1–29, Jan. 2023.

[14] Tieming Chen, Xuebo Qiu, Zhengqiu Weng, Tiantian Zhu, Mingqi Lv, Keda Sun. "Security and Communication Networks Research Article Open Access HTTPSmell: A Deep Learning Approach on Malicious HTTP Traffic Detection via Data Augmentation and Label Refactoring". 06 November 2024.

[15] Dash, N., Chakravarty, S., Rath, A.K. et al. An optimized LSTM-based deep learning model for anomaly network intrusion detection. Sci Rep 15, 1554 (2025).

[16] Ramu, C., Rao, T.S. & Rao, E.U.S. Attack classification in network intrusion detection system based on optimization strategy and deep learning methodology. Multimed Tools Appl 83, 75533–75555 (2024).

[17] N. Moustafa and J. Slay, ``UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS), Nov. 2015, pp. 16.

[18] **W. Li, Y. Fan and Q. Xu, "Teaching-Learning-Based Optimization Enhanced With Multiobjective Sorting Based and Cooperative Learning," in IEEE Access, vol. 8, pp. 65923-65937, 2020**

[19] **Wu, X.; Li, S.; Wu, F.; Jiang, X. Teaching–Learning Optimization Algorithm Based on the Cadre–Mass Relationship with Tutor Mechanism for Solving Complex Optimization Problems. Biomimetics 2023, 8, 462.**

[20] D. P. Truong, E. Skau, D. Desantis and B. Alexandrov, "Boolean Matrix Factorization via Nonnegative Auxiliary Optimization," in IEEE Access, vol. 9, pp. 117169-117177, 2021.

[21] W. Zhu and Y. Yan, "Joint Linear Regression and Nonnegative Matrix Factorization Based on Self-Organized Graph for Image Clustering and Classification," in IEEE Access, vol. 6, pp. 38820-38834, 2018.

[22] Kuo, C.-T.; Lin, J.-J.; Jen, K.-K.; Hsu, W.-L.; Wang, F.-C.; Tsao, T.-C.; Yen, J.-Y. Human Posture Transition-Time Detection Based upon Inertial Measurement Unit and Long Short-Term Memory Neural Networks. Biomimetics 2023, 8, 471.

[23] https://github.com/defcom17/NSL_KDD/blob/master/Original%20NSL%20KDD%20Zip.zip