An Enhanced Network Intrusion Detection System Using ADASYN and Hybrid Residual Block Techniques

Harish G N
Department of Computer Science & Engineering, Sri Siddhartha Academy of Higher Education, Tumakuru, Karnataka, India
acchu.harishgn@gmail.com (corresponding author)

Annapurna H S
Department of Information Science & Engineering, Sri Siddhartha Institute of Technology, Tumakuru, Karnataka, India
annapoornahs@ssit.edu.in

Received: 15 March 2025 | Revised: 11 April 2025, 23 April 2025, and 27 April 2025 | Accepted: 1 May 2025

 DOI: https://doi.org/10.48084/etasr.10961

ABSTRACT

Network security relies on Intrusion Detection Systems (IDS), but current models have problems with feature extraction and inaccurate classification, particularly on unbalanced and small datasets, leading to less effective detection of attack traffic patterns in actual situations. This study suggests an adaptive synthesis method based on an Enhanced Residual Network (ERN) to address these problems to facilitate IDS models in learning from sparse and unbalanced data and increase their detection performance. For optimal distribution of training data and enhanced feature representation, the proposed ERN uses the Inception-ResNet architecture in conjunction with custom sampling modules. Oversampling techniques are used to balance the dataset. The model is trained, tested, and compared to traditional deep learning methods. The proposed model outperforms conventional ones in terms of accuracy, dependability, and feature extraction capacity, as experimentally shown by an intrusion detection accuracy ranging from 89.40% to 91.88%. These results show that the proposed method is a solid choice for tough data settings looking to enhance intrusion detection.

Keywords-unstable data connection; residual neural network; adaptive synthesis; intrusion detection

I. INTRODUCTION

Network security plays an important role in providing safety measures such as firewalls, encryption, authentication systems, etc., as traditional methods and technologies have failed to prevent attacks. Denial of Service (DoS), Remote to Local (R2L), User to Root (U2R), and probing are examples of attacks. Intrusion Detection Systems (IDS) classify attacks based on specific attributes [1], helping in the early stage of attack detection to provide the necessary preventive measures. In [2, 3], the use of conventional machine learning methods was described for the detection of attacks, including data collection, feature selection, and classification. Different levels of encoding strategies discretize the attributes to improve the performance of the model. Parameters such as accuracy and efficiency are affected by different feature selection and analysis methods.

In [4], a ResNet-CNN model was presented to increase IDS performance, achieving 98.94% (binary) and 98.92% (multiclass) accuracy. However, overfitting, dataset dependence, and high computational cost are some of its drawbacks. In [5], ResIncepNet-SA was introduced, which combined CNN, InceptionNet, ResNet, and self-attention. This method used PCA-ADASYN to address feature extraction and dataset imbalance. Although this model was efficient, it had overfitting issues, high resource consumption, and lacked real-time validation. In [1], samples were balanced and feature learning was enhanced by utilizing ADASYN and an enhanced CNN (AS-CNN). Although this model performed better than CNN/RNN models, it is limited due to resource utilization and dataset specificity. In [2], a hybrid attention mechanism was proposed for IDS was proposed, achieving high accuracy but facing generalization problems and overfitting risks in complex networks. In [3], different IDS models were evaluated on the UNSW-NB15 dataset, with an emphasis on false alarms and accuracy. In [6], the Res-Net-TranBiLSTM model extracted spatial-temporal features for IDS, achieving up to 99.56% accuracy, but with generalization and real-time deployment issues. In [1], ADASYN and SPC-CNN were combined to improve detection rates and accuracy compared to conventional CNN/RNNs. Two major drawbacks are real-time deployment

and dataset dependency. In [7], a 97% accurate CNN-based NIDS for WSNs was presented, but additional testing is required for scalability and generalizability. KNN was used in [8] to detect IoT intrusions, achieving 99.99% accuracy, but its wider use is restricted by feature generalization and dataset dependence. In [9], a DNN-based IDS was developed for wireless networks, achieving an accuracy of ~92% but having drawbacks with overfitting and reliance on a specific dataset. In [10], ADASYN was used in conjunction with Random Forest on CICIDS 2017, but the evaluation of a single dataset limits it.

Outperforming other GAN models, the VAE-WACGAN model [11] produced synthetic minority samples for improved detection but had problems with diversity and scalability. Deep learning-based IDS techniques were reviewed in [12]. In [13], 119 important papers on anomaly-based NIDS were reviewed, offering recommendations for future research. With an accuracy of more than 98%, the IGAN model [14] enhanced minority class detection through Lenet 5 and LSTM, but contextual constraints and overfitting continued to exist. In [15], an SVM-deep learning model was used to detect social media intrusions but had issues with generalization and resource requirements. DLHA [16] achieved high detection rates using SVM and Naive Bayes, but it might not adapt well to new attack types. In [17], a lightweight hybrid CNN+LSTM was proposed for IoT-IDS, achieving ~98% across evaluation metrics, but its effectiveness against more general threats must be confirmed.

II. PROPOSED METHOD

A. Adaptive Synthetic Sampling (ADASYN)

ADASYN, an advancement of the SMOTE algorithm, is a sampling process for unbalanced datasets [18]. To improve the identification of positives, ADASYN attempts to produce more synthetic cases in the class with fewer positive instances. This approach creates a distribution function by counting the number of instances that are negative neighbors in each positive instance's K-Nearest Neighbors (KNN). The probability distribution function determines the synthetic instances created from the positive one [7]. When minority samples are harder to learn than easier minority sample classes, ADASYN generates more points of information and observations. The fact that ADASYN generates a set number of instances for each minority instance using weighted distributions of its neighbors makes it ultimately a pseudo-probabilistic algorithm. While ADASYN utilizes a density distribution as an indicator to automatically assess the number of synthetic data samples to be created for every case of the minority sample group, SMOTE gives each minority sample class the same chance of being picked during the development of the artificial information samples.

Thus, by shifting the classification determination boundary to the difficult cases and mitigating the bias caused by the class disparity, the ADASYN technique enhances data distribution. It is best to deal with aberrations during data preprocessing before using the ADASYN technique since it is highly sensitive to outliers. By producing artificial data points for these challenging-to-grasp samples, ADASYN, in contrast to SMOTE, focuses more on the minority specimens that are challenging to learn.

```
Algorithm 1: ADASYN
Input: m samples in the training set where
       i=1,2,3,4,…,m
       nl+ns=m where
       nl=Large samples
       ns=Small samples
Step 1: Imbalance degree calculation:
        d= ns/nl
Step 2: Calculation of total number of
        samples that are needed for
        synthesization
        Q=β(nl−ns)
Step 3: Calculate Ri using
        Ri=γ/k
Step 4: Normalize R using
        Ri'=Ri/(∑_l^Ns Ri)
Step 5: Calculate the total number of
        samples to be generated
        gi = Ri'*G,
        here G is given as the number of
        samples to be synthesized
Output: The new dataset
```

B. Residual Networks (ResNets)

Residual blocks are used to solve different types of gradient problems. The residual block uses the skip connection technique, which works based on the connection of activation layers to the next layers by skipping some unwanted layers to form a residual structure block, as shown in Figure 1. ResNet works based on the network fitting residual mapping rather than learning the mapping. The network fit formula is:

$$F(x) = H(x) - x \quad (1)$$

which gives:

$$H(x) = F(x) + x \quad (2)$$

The main purpose of the skip connection network is to restrict layers that degrade the performance of the model. Regularization is used to skip layers, which also helps to train the neural network without any other negative disturbances. This skip network has the set of a 34-layer plain network used in the ResNet.

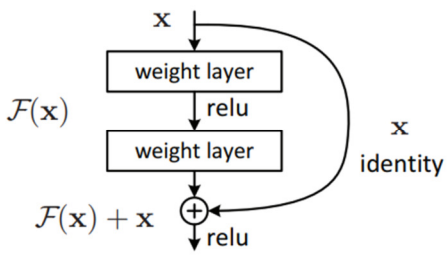

Fig. 1. Skip connection.

III. IMPLEMENTATION

A. Method of Hybrid Sampling Integrating RENN with ADASYN

This study proposes a hybrid sampling strategy that combines ADASYN and RENN. The basic concept involves creating two sets of samples, one for the majority class and one for the minority class. The clustering method is then employed to filter out noise from the new sample set. Subsequently, the two datasets are merged to construct a balanced dataset. The larger and smaller samples are merged to build a dataset where the imbalance is eliminated.

The RENN method is utilized to represent the majority class, while the ADASYN-based algorithm is employed to augment minority samples. A unique approach is adopted in this combined sampling strategy, wherein the algorithm takes the initial sample from the minority class (P), along with the desired number of samples and their proportions. The algorithm generates two sets of examples: one containing a pair from the majority class (newN) and the other containing an equal number of examples from the minority class (newP). The following basic steps are involved in the sampling strategy:

1. Assess the degree of imbalance in the dataset.
2. Calculate the proportion by identifying each sample in the majority class (N) and determining its $k1$ nearest neighbors. Normalize the proportions and determine the total number of samples to synthesize for each single sample.
3. Use the generation of gi samples for each sample in the majority class (N) to create a new group of minority-level samples.
4. For each sample in the minority class (P), select $k2$ nearest neighbors from newN.
5. Determine the number of minority samples within the $k2$ nearest neighbors of each sample and remove the sample if the total exceeds a specified threshold (e.g., $e = 1$).
6. Repeat steps 4 and 5 to create a new sample collection for the majority class.
7. Deduct the samples from the new P and new N to obtain the final new N and P sets.

B. Residual Learning

Let x be the input data on top of this layer and let $H(x)$ be the basic translation corresponding to several layers (not necessarily full mesh). The assumption that the nonlinear layers can simultaneously approach complex behavior is the same as the assumption that each layer can quickly compare the remaining processes, or $H(x)$-x, indicating that the inputs and outputs are of the same size. Therefore, the remaining constant $F(x)$=$H(x)$-x to converge is allowed to converge, instead of expecting $H(x)$ to be a similar layer. The original function is $F(x)$+x. The ease of learning for each type can be different, although both are expected to quickly approach the objective function (such as the hypothesis).

This recovery taps into the paradoxical nature of the problem of degradation. As stated above, if the additional layer can be built as an identity map, the more complex model should have smaller errors during training compared to its narrower partner. The problem of corruption means that solvers can have difficulty solving similar maps using many linear layers. In this training reform, in the scenario where the identity translation is optimal, the identity map can be efficiently approximated by reducing the weight of the nonlinear layer to zero. Although identity mapping is not optimal in real-world situations, this reform can help prevent this problem. The solver can more easily detect disturbances compared to the identity projection, rather than learning the function as a new one, especially if the optimal function is more closely related to the identity map than to the null mapping. Empirical evidence shows that the learning residual function typically produces a small response, indicating that the identity map offers effective conditions.

C. Residual Network

Evaluations were performed on 18-layer and 34-layer ResNets. The basic design is exactly the same as the direct network mentioned above, including a short connection for each pair of 3×3 filters. Unlike its simple counterparts, there are no additional features. After repeated training, the 34-layer ResNet outperforms the 18-layer ResNet by about 2.8%. In addition, the 34-layer ResNet shows improved evaluation results, indicating that the degradation problem is effectively solved in this context and allows for the use of a deeper architecture for better accuracy. Furthermore, due to the effective reduction in training error, the 34-layer ResNet reached a maximum error of 3.5%, showing a significant improvement compared to its simpler counterpart.

D. Evaluation Parameters

True Positive (TP) denotes the total number of attacks determined by the system that are correct. False Negative (FN) is the total number of attacks that were not detected. True Negative (TN) denotes the normal instances detected, and False Positive denotes the number of normal instances detected as attacks. These variables are used to calculate different performance parameters to analyze performance. Accuracy is given by:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (3)$$

Recall is the ratio of total positive samples and to the sum of TP and FN:

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (4)$$

Precision is the ratio of TP to the sum of TP and FP:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (5)$$

F1-score is the harmonic average of precision and recall:

$$\text{F1} - \text{score} = \frac{2\times \text{Precision}\times \text{Recall}}{\text{Precision+Recall}} \quad (6)$$

E. Dataset

The UNSW-NB15 dataset, created in 2015 by the Australian Cyber Security Center (ACCS) Cyberspace Intelligence Laboratory, was generated using IXIA's PerfectStorm software to emulate real-world cyber environments [18-22]. Comprising 47 features categorized into two groups, the dataset encompasses nine attack methods, including spoofs, probes, Trojans, DoS, generic, exploration, shellcode, and worms. The performance of the model in this task was assessed using the initial training and test sets.

TABLE I. UNSW-NB15 DATASET

Category	Test	Train
Normal	5600	37005
Backdoor	1746	583
Analysis	2000	677
Fuzzers	18185	6062
Shell code	1133	378
Reconnaissance	10492	3496
Exploits	33393	11132
DoS	12264	4089
Worms	130	44
Generic	40000	18871
Total	175343	82337

TABLE II. THE EXPERIMENTAL ENVIRONMENT

Name	Model/Version
GPU	NVIDIA GetForceRTX3060
CPU	Intel i7-12700HZ
RAM	32 GB
Language	Python 3.10
Pycharm	2023.3.3

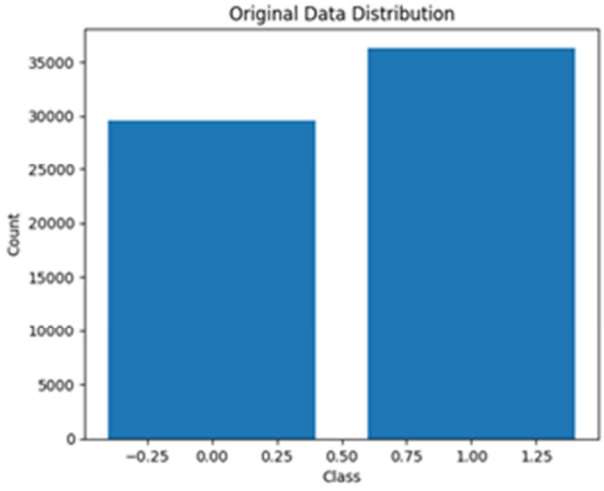

Fig. 2. Data before ADASYN application.

IV. RESULTS AND DISCUSSION

A. Data Preprocessing

The preprocessing steps, described in Figures 3-5, include the following steps:

- Converting the original non-numerical features into numerical and normalizing them
- Sampling using an algorithm that combines ADASYN and RENN
- Using feature selection
- Convert the resulting data into a grey-scale map

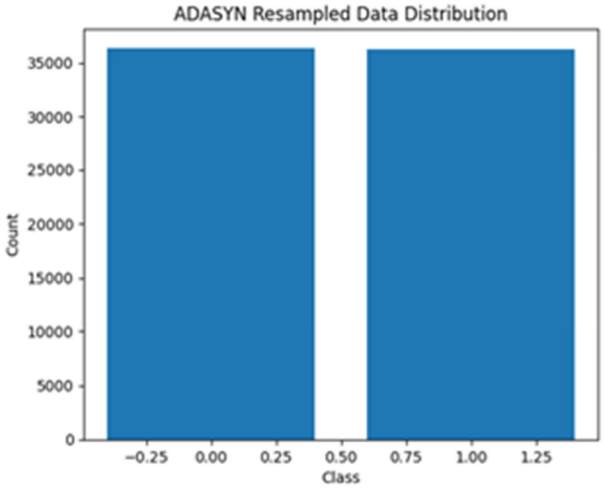

Fig. 3. Balanced data after ADASYN application.

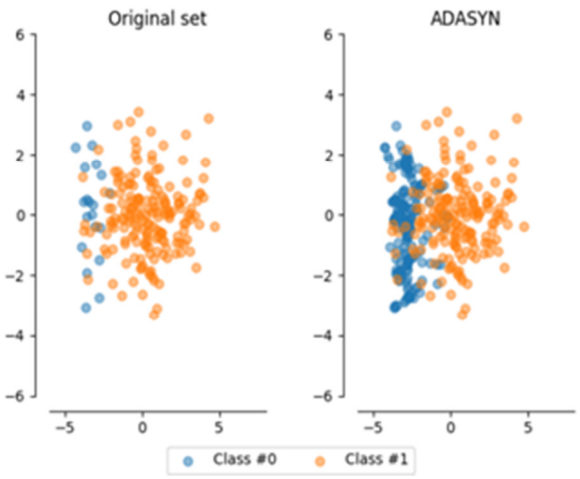

Fig. 4. Schematic diagram of ADASYN application.

B. Attack Detection

Figure 6 shows the confusion matrix corresponding to the ResNet neural network modeling method. The proposed model has better performance compared to the traditional algorithms, with considerably higher accuracy, precision, recall, and F1 scores. As shown in Figure 7, better performance results were achieved due to ADASYN, which does not ignore the minority samples and the performance of the network does not degrade even after adding more data due to residual blocks.

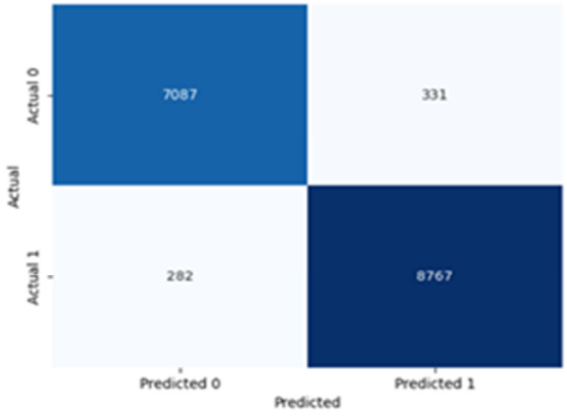

Fig. 5. Confusion matrix.

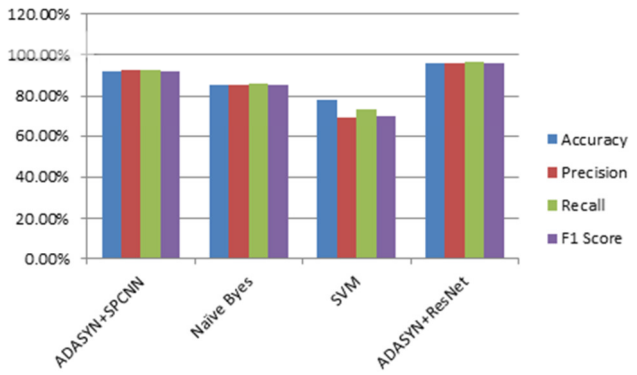

Fig. 6. Comparison of the proposed method with traditional algorithms.

V. CONCLUSION

By addressing undesired distribution and redundant information across multiple classes, the study seeks to enhance the performance of neural network-based IDS. The ADASYN technique is used to balance class distributions, reduce bias towards majority classes, and enhance the detection of minority class intrusions. The split-based ResNet architecture enables interchannel redundancy mitigation and multiscale feature extraction. The hybrid model, which combines ADASYN and ResNet, exhibits notable gains in multiclass intrusion detection tasks, including better classification performance, lower false alarm rates, and greater detection accuracy, particularly for minority classes. Future research should focus on increasing the detection accuracy and maximizing computational efficiency for small sample intrusions in real-time network environments.

REFERENCES

[1] Z. Hu, L. Wang, L. Qi, Y. Li, and W. Yang, "A Novel Wireless Network Intrusion Detection Method Based on Adaptive Synthetic Sampling and an Improved Convolutional Neural Network," *IEEE Access*, vol. 8, pp. 195741–195751, 2020, https://doi.org/10.1109/ACCESS.2020.3034015.

[2] X. Hu, X. Meng, S. Liu, and L. Liang, "An Improved Algorithm for Network Intrusion Detection Based on Deep Residual Networks," *IEEE Access*, vol. 12, pp. 66432–66441, 2024, https://doi.org/10.1109/ACCESS.2024.3398007.

[3] K. Kotecha *et al.*, "Enhanced Network Intrusion Detection System," *Sensors*, vol. 21, no. 23, Jan. 2021, Art. no. 7835, https://doi.org/10.3390/s21237835.

[4] S. Farhan, J. Mubashir, Y. U. Haq, T. Mahmood, and A. Rehman, "Enhancing network security: an intrusion detection system using residual network-based convolutional neural network," *Cluster Computing*, vol. 28, no. 4, Feb. 2025, Art. no. 251, https://doi.org/10.1007/s10586-025-05156-9.

[5] G. Liu, T. Zhang, H. Dai, X. Cheng, and D. Yang, "ResInceptNet-SA: A Network Traffic Intrusion Detection Model Fusing Feature Selection and Balanced Datasets," *Applied Sciences*, vol. 15, no. 2, Jan. 2025, Art. no. 956, https://doi.org/10.3390/app15020956.

[6] S. Wang, W. Xu, and Y. Liu, "Res-TranBiLSTM: An intelligent approach for intrusion detection in the Internet of Things," *Computer Networks*, vol. 235, Nov. 2023, Art. no. 109982, https://doi.org/10.1016/j.comnet.2023.109982.

[7] H. Sadia *et al.*, "Intrusion Detection System for Wireless Sensor Networks: A Machine Learning Based Approach," *IEEE Access*, vol. 12, pp. 52565–52582, 2024, https://doi.org/10.1109/ACCESS.2024.3380014.

[8] M. Mohy-Eddine, A. Guezzaz, S. Benkirane, and M. Azrour, "An Intrusion Detection Model using election-Based Feature Selection and K-NN," *Microprocessors and Microsystems*, Oct. 2023, Art. no. 104966, https://doi.org/10.1016/j.micpro.2023.104966.

[9] M. Rajkumar, V. S. Lakshi, R. Karthik, and S. Pavithra, "Optimized Deep Learning Mechanism for Intrusion Detection: Leveraging RFE-Based Feature Selection and PCA for Improved Accuracy," in *2024 International Conference on Sustainable Communication Networks and Application (ICSCNA)*, Theni, India, Dec. 2024, pp. 1517–1522, https://doi.org/10.1109/ICSCNA63714.2024.10863936.

[10] Z. Chen, L. Zhou, and W. Yu, "ADASYN–Random Forest Based Intrusion Detection Model," in *2021 4th International Conference on Signal Processing and Machine Learning*, Beijing, China, Aug. 2021, pp. 152–159, https://doi.org/10.1145/3483207.3483232.

[11] W. Tian, Y. Shen, N. Guo, J. Yuan, and Y. Yang, "VAE-WACGAN: An Improved Data Augmentation Method Based on VAEGAN for Intrusion Detection," *Sensors*, vol. 24, no. 18, Sep. 2024, Art. no. 6035, https://doi.org/10.3390/s24186035.

[12] J. Lansky *et al.*, "Deep Learning-Based Intrusion Detection Systems: A Systematic Review," *IEEE Access*, vol. 9, pp. 101574–101599, 2021, https://doi.org/10.1109/ACCESS.2021.3097247.

[13] Z. Yang *et al.*, "A systematic literature review of methods and datasets for anomaly-based network intrusion detection," *Computers & Security*, vol. 116, May 2022, Art. no. 102675, https://doi.org/10.1016/j.cose.2022.102675.

[14] Y. N. Rao and K. Suresh Babu, "An Imbalanced Generative Adversarial Network-Based Approach for Network Intrusion Detection in an Imbalanced Dataset," *Sensors*, vol. 23, no. 1, Jan. 2023, Art. no. 550, https://doi.org/10.3390/s23010550.

[15] K. M. Abuali, L. Nissirat, and A. Al-Samawi, "Advancing Network Security with AI: SVM-Based Deep Learning for Intrusion Detection," *Sensors*, vol. 23, no. 21, Jan. 2023, Art. no. 8959, https://doi.org/10.3390/s23218959.

[16] T. Wisanwanichthan and M. Thammawichai, "A Double-Layered Hybrid Approach for Network Intrusion Detection System Using Combined Naive Bayes and SVM," *IEEE Access*, vol. 9, pp. 138432–138450, 2021, https://doi.org/10.1109/ACCESS.2021.3118573.

[17] R. H. Altaie and H. K. Hoomod, "An Intrusion Detection System using a Hybrid Lightweight Deep Learning Algorithm," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 16740–16743, Oct. 2024, https://doi.org/10.48084/etasr.7657.

[18] N. Moustafa, J. Slay, and G. Creech, "Novel Geometric Area Analysis Technique for Anomaly Detection Using Trapezoidal Area Estimation on Large-Scale Networks," *IEEE Transactions on Big Data*, vol. 5, no. 4, pp. 481–494, Sep. 2019, https://doi.org/10.1109/TBDATA.2017.2715166.

[19] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia, Nov. 2015, pp. 1–6, https://doi.org/10.1109/MilCIS.2015.7348942.

[20] N. Moustafa and J. and Slay, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Information Security Journal: A Global Perspective*, vol. 25, no. 1–3, pp. 18–31, Apr. 2016, https://doi.org/10.1080/19393555.2015.1125974.

[21] N. Moustafa, G. Creech, and J. Slay, "Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models," in *Data Analytics and Decision Support for Cybersecurity: Trends, Methodologies and Applications*, I. Palomares Carrascosa, H. K. Kalutarage, and Y. Huang, Eds. Springer International Publishing, 2017, pp. 127–156.

[22] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems," in *Big Data Technologies and Applications*, 2021, pp. 117–135, https://doi.org/10.1007/978-3-030-72802-1_9.