Vol. 3, No. 3, Desember 2022, hal. 242-248

Jurnal Computer Science and Information Technology (CoSciTech)

p-ISSN: 2723-567X
e-ISSN: 2723-5661

http://ejurnal.umri.ac.id/index.php/coscitech/index

# Implementasi adasyn untuk imbalance data pada dataset UNSW-NB15

**Januar Al Amien[*1], Yoze Rizki [2], Mukhlis Ali Rahman Nasution[3]**
Email: [1]januaralamien@umri.ac.id, [2] yozerizki@umri.ac.id, [3]170401162@student.umri.ac.id

[123]Teknik Informatika, Fakultas Ilmu Komputer, Universitas Muhammadiyah Riau

Diterima: 30 November 2022 | Direvisi: - | Disetujui: 16 Desember 2022
©2020 Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Muhammadiyah Riau, Indonesia

**Abstrak**

Di masa Machine Learning pada saat ini, para peneliti bekerja keras untuk mengembangkan algoritma yang meningkatkan kemungkinan prediksi yang benar dengan akurasi yang lebih baik. Data tidak seimbang adalah ketika ukuran sampel dari satu kelas jauh lebih besar dari kelas lain, sampel minoritas dapat diperlakukan sebagai noise dalam proses klasifikasi, yang mengakibatkan hasil algoritma klasifikasi yang tidak memuaskan. Pada penelitian ini peneliti menggunakan dataset UNSW-NB15, setelah menggabungkan data train dan test, terdapat data tidak seimbangan pada kelas label, yaitu 164673 untuk label 1 dan 93000 untuk label 0. Tujuan penelitian ini untuk mengatasi masalah ketidakseimbangan data pada binary class dengan menggunakan teknik ADASYN dan mendeteksi serangan malware pada dataset UNSW-NB15 dengan menerapkan model algoritma Random Forest dan teknik ADASYN agar mendapatkan performa yang cukup baik. Berdasarkan hasil pengujian dengan teknik ADASYN untuk penanganan ketidakseimbangan data pada Binarry Class dan menggunakan model algoritma Random Forest, serta Hyperparameter Optuna untuk klasifikasi Anomali pada data UNSW-NB15 memperoleh akurasi yang cukup baik. Pada beberapa split data mendapatkan nilai akurasi tertinggi pada split data 90/10 dengan hasil 99.86%. dari segi waktu tercepat didapat pada split data 60/40 yaitu 1,85 seconds.

**Kata kunci**: *Machine learning, ADASYN, Random Forest, Optuna, UNSW-NB15.*

## *Adasyn implementation for imbalance data on UNSW-NB15 dataset*

***Abstract***

*In today's era of Machine Learning, researchers are working hard to develop yahoo that increases the probability of correct predictions with better accuracy. Unbalanced data is when the sample size of one class is much larger than that of another class, minority samples may be treated as noise in the classification process, resulting in unsatisfactory classification results. In this study the researchers used the UNSW-NB15 dataset, after combining the train and test data, there were unbalanced data in the label class, namely 164673 for label 1 and 93000 for label 0. The purpose of this study was to overcome the problem of data imbalance in the binary class by using the ADASYN technique and detecting malware attacks on the UNSW-NB15 dataset by applying the Random Forest algorithm model and the ADASYN technique in order to get a fairly good performance. Based on the results of testing with the ADASYN technique for handling data imbalances in the Binarry Class and using the Random Forest algorithm model, as well as the Optuna Hyperparameter for the classification of anomalies in UNSW-NB15 data, the accuracy is quite good. In some data splits, the highest accuracy value is 90/10 data split with 99.86% results. in terms of the fastest time obtained on the 60/40 data split, which is 1.85 seconds.*

**Keywords**: *Machine learning, ADASYN, Random Forest, Optuna, UNSW-NB15.*

## 1. PENDAHULUAN

Saat ini, banyak perusahaan yang tertarik dengan teknologi layanan (yaitu sistem) untuk menyelesaikan proses yang lebih cepat daripada cara tradisional, dan agar sistem ini lebih efisien, harus dilindungi dari ancaman dan informasi keamanan harus tetap terjaga [1]. Keamanan jaringan adalah suatu ilmu komprehensif yang melibatkan teknologi komputer, teknologi jaringan, teknologi komunikasi, kriptografi, teknologi keamanan informasi [2].

Malware berasal dari kata malicious dan software dapat diartikan perangkat lunak yang digunakan untuk melakukan perusakan sistem, pencurian atau pengumpulan informasi, hingga mendapatkan akses terhadap suatu sistem. Terdapat beberapa cara penyebaran malware seperti email phising, serangan rekayasa sosial, dan downloader. Tujuan dari penyebaran malware adalah untuk pencurian data rahasia, pengumpulan informasi seperti password dan email, serta spamming [3].

IDS adalah sebuah aplikasi perangkat keras atau perangkat lunak yang otomatis bekerja untuk memonitor kejadian pada sebuah jaringan komputer dan sekaligus menganalisis masalah keamanan jaringan. Tujuan IDS adalah memonitoring aset jaringan sehingga dapat mendeteksi perilaku yang tidak lazim, atau kegiatan yang tidak sesuai [4].

Machine learning memungkinkan komputer belajar memecahkan masalah tertentu dan membuat prediksi berdasarkan pengamatan masa lalu. Algoritma machine learning sangat bervariasi, dan dapat dikelompokkan berdasarkan teknik pembelajaran, kesamaan tugas dalam menjalankan fungsi, atau kedalaman pembelajaran. Machine learning mempunyai tiga strategi, yaitu Supervised Learning, Unsupervised Learning dan Semi Supervised Learning [5].

Penelitian ini menggunakan dataset UNSWNB15, dengan menggabungkan data train dan test, ditemukan data tidak seimbangan pada kelas label, yaitu 164673 untuk label 1 dan 93000 untuk label 0. Kumpulan data ini bisa di akses serta di download untuk tujuan penelitian dan dapat diakses dari tautan https://research.unsw.edu.au/projects/unsw-nb15-dataset. Jumlah total record adalah 257.673, disimpan dalam empat file CSV. Selanjutnya, partisi dari dataset ini dikonfigurasi sebagai set pelatihan standar dan set pengujian. Jumlah catatan dalam set pelatihan adalah 175.341 record, dan set pengujian adalah 82.332 record [6].

Data tidak seimbang adalah Ketika ukuran sampel dari satu kelas jauh lebih besar dari kelas lain, sampel minoritas dapat diperlakukan sebagai noise dalam proses klasifikasi, yang mengakibatkan hasil algoritma klasifikasi yang tidak memuaskan. Data tidak seimbang dapat diatasi dengan cara menggunakan metode over sampling untuk mensintesis sampel minoritas dan menggunakan metode under sampling untuk mengurangi sampel mayoritas, sehingga dapat memecahkan masalah ketidakseimbangan kelas [7].

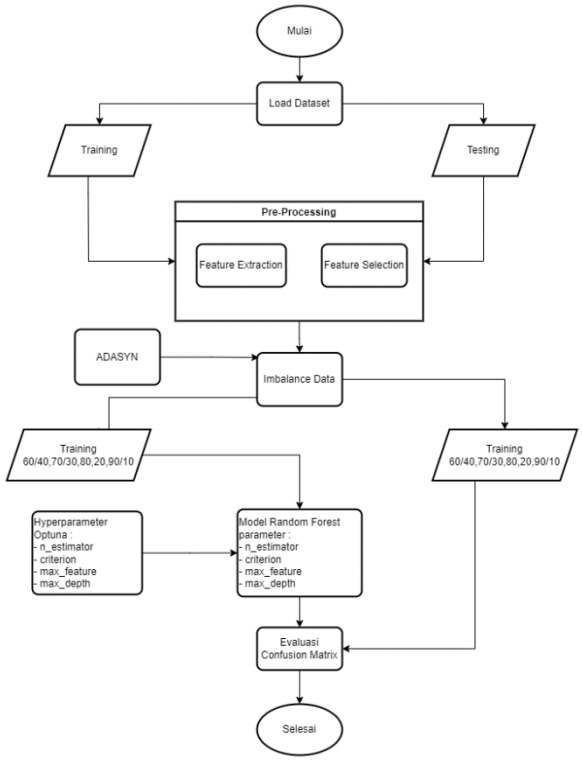
Sebab terdapat nya imbalance pada dataset UNSW-NB15, penulis menggunakan teknik ADASYN untuk menyeimbangkan data pada kelas yang imbalance. beberapa ahli telah memperkenalkan algoritma over-sampling SMOTE ke dalam bidang intrusion detection system. Namun, ketika algoritma ini menghasilkan sampel minoritas, SMOTE tidak mempertimbangkan lingkungan sekitar dari titik sampel minoritas. Kinerja teknik over sampling secara keseluruhan lebih baik. Di antara teknik over sampling, kinerja ADASYN relatif lebih baik [8].

Dalam penelitian ini, algoritma yang digunakan ialah Random Forest, Random Forest termasuk algorithma klasifikasi berbasis bagging essemble yang popular. Random Forest terdiri atas banyak pohon keputusan (decission tree). Dan memiliki kelebihan yaitu menghasilkan error yang rendah, menghasilkan klasifikasi yang baik, dan dapat mengatasi data training yang sangat besar serta efektif untuk mengestimasi missing data, dalam beberapa penelitian deteksi serangan algorithma Random Forest menunjukkan kinerja yang baik [9]. Pada penelitian [10] melakukan penelitian dengan dataset yang diperoleh dari 30.718 user perusahaan pada April 2019 menggunakan algoritma Random Forest dengan teknik SMOTE dan ADASYN untuk imbalance data. Hasil menunjukkan Random Forest dengan menggunakan teknik ADASYN lebih unggul dari pada menggunakan teknik SMOTE. Pada beberapa penelitian dengan menggunakan algoritma Random Fores dan dataset yang sama yaitu UNSW-NB15, menggunakan teknik SMOTE [11] mendapatkan akurasi 95,1%, dengan teknik clustering under-sampling [12] mendapatkan akurasi 97,1%, dengan teknik Random Over-sampling [13] mendapatkan akurasi 87,82%, dan menggunakan algoritma Random Fores saja [14] mendapatkan hasil akurasi 98,86%.

Tujuan penelitian ini adalah untuk memperoleh akurasi yang cukup baik dengan melakukan penanganan imbalance pada dataset.

## 2. METODE PENELITIAN

Kerangka kerja ini menggunakan subset dari dataset UNSW-NB15. Penelitian ini terdiri dari dua langkah utama. Langkah pertama melibatkan *pre-processing* data, di mana *feature selection* dan *extraction* dilakukan. Karena sifat dimensi yang tinggi dari kumpulan data, beberapa fitur yang tidak relevan atau berlebihan dapat menyebabkan penurunan akurasi deteksi serangan. Untuk mengatasi masalah ini, pemilihan fitur digunakan, di mana hanya subset fitur *numeric* yang dipilih. Setelah itu, kami kemudian membahas masalah *imbalance class*. Pada langkah selanjutnya melakukan proses pengklasifikasian dengan menggunakan *hyperparameter* optuna untuk mendapatkan akurasi maksimum. Terakhir, *accuracy, precision, recall.*




Jelaskan metode penelitian dan teknik penelitian yang digunakan. Jelaskan dengan ringkas, tetapi tetap akurat seperti ukuran, volume, replikasi dan teknik pengerjaan. Untuk metode baru harus dijelaskan secara rinci agar peneliti lain dapat mereproduksi percobaan.

2.1. Dataset

Kumpulan data UNSW-NB 15 dibuat oleh alat *IXIA PerfectStorm* di Lab *Cyber Range* dari Pusat Keamanan *Australian Center for Cyber Security* (ACCS) untuk menghasilkan campuran aktivitas normal modern yang nyata dan perilaku serangan kontemporer sintetis [15]. Alat IXIA digunakan sebagai generator lalu lintas serangan bersama dengan lalu lintas normal, perilaku serangan itu dipelihara dari situs CVE untuk tujuan representasi nyata dari lingkungan ancaman modern. Kumpulan data diberi label kategori serangan yaitu, *attack_cat* dan label. untuk *record* normal diberi label 0 (nol) dan 1 (satu) untuk serangan. Lalu serangan diklasifikasikan menjadi sembilan kelompok [16].

Tabel 1 Jenis-jenis serangan

| ***Tipe*** | ***Number*** |
|---|---|
| *Fuzzers* | 24,246 |
| *Analisis* | 2,677 |
| *Backdoor* | 2,329 |
| *DoS* | 16,353 |
| *Eksploitasi* | 44,525 |
| *Generik* | 215,481 |
| *Reconnaissance* | 13,987 |
| *Shellcode* | 1,511 |
| *Worm* | 174 |

2.2. Feature Extraction

Feature Extraction adalah mengekstrak sebuah fitur yang akan kita gunakan pada tahap selanjutnya. Dalam penelitian ini akan dilakukan feature extraction terhadap binary class, yaitu terdapat dua *class* yang jumlah nya tidak seimbang, dimana *class* 1 terdapat 164.673 dan *class* 0 terdapat 93.000.

2.3. Feature Selection

*Feature Selection* adalah proses mengidentifikasi beberapa *feature* terpenting yang membantu mendapatkan akurasi model yang lebih baik. Pada tahapan ini dilakukan pembersihan *feature* yang tidak digunakan dalam klasifikasi dan hanya memilih features yang numeric.

2.4. ADASYN

ADASYN adalah versi perbaikan dari Sintetis Minoritas *Over- Sampling Technique (SMOTE),* yang digunakan untuk menghindari *overfitting* yang terjadi ketika replika yang tepat dari instance minoritas ditambahkan ke dataset utama. Ide utama dari algoritma ADASYN adalah menggunakan distribusi densitas sebagai kriteria untuk secara otomatis menentukan jumlah sampel sintetis yang sesuai yang perlu dihasilkan untuk setiap contoh pada data minoritas [17].

Algorithm *ADASYN*

*The data set D_tr of m samples can be expressed as { x_i , y_i}, where i = 1 , . . . , m, x_i is the sample of the n-dimensional feature space X , and y_i is the label of the sample x_i, y_i Y = {−1, 1}. The number of majority samples is m_l, and the number of minority samples is m_s.*

*(1)* *Calculate the minority ratio for the majority example using :*

$$d = m_s / m_l \quad (1)$$

*Where* $d \in [0, 1]$

*(2)* *If* $d < d_{th}$ *then (*$d_{th}$ *is a predefined threshold for the maximum tolerable level of class imbalance ratio) :*

*(a)* *Count the number of synthetic data samples that need to be generated for the minority class:*

$$G = (m_l - m_s) \times \beta \quad (2)$$

*Where* $\beta \in [0, 1]$ *is a parameter used to determine the desired level of equilibrium after the creation of synthetic data.* $\beta$*=1 means a completely balanced data set is created after the generalization process.*

*(b)* *For each example* $x_i$ *∈ minority class, find K nearest neighbor based on Euclidean distance in dimensional space n, and count the rasio which is defined as* $r_i$*:*

$$r_i = \Delta_i / K \quad (3)$$

$\Delta_i$ *is the number belonging to the majority class at K The closest neighbor of* $x_i$*, so* $r_i \in [0, 1]$*.*

*(c)* *Normalization* $\hat{r}_i = r_i / \sum_{i=1}^{m_s} r_i$*, so* $r_i$ *is the density distribution (* $\sum_i r_i = 1$ *)*

*(d)* *Calculate the number of synthetic data samples that need to be generated for each minority sample* $x_i$ *:*

$$g_i = \hat{r}_i \times G \quad (4)$$

*where G is the total number of synthetic data samples that need to be generated for the minority class as defined in Eq (2).*

*(e)* *For each sample of minority class data* $x_i$*, create an example of synthetic data* $g_i$ *according to the following steps :*

*Do* ***Loop*** *from 1 to* $g_i$ *:*

*(i)* *Randomly select one sample of minority data,* $x_{zi}$*, from K nearest neighbor for data* $x_i$*.*

*(ii)* *Generate synthetic data samples:*

$$s_i = x_i + (x_{zi} - x_i) \times \lambda \quad (5)$$

*where* $(x_{zi} - x_i)$ *is the difference vector in the n-dimensional space, and λ is a random number:* $\lambda \in [0, 1]$

*End* ***Loop***

2.5. Random Forest

*Random Forest* termasuk algoritma klasifikasi berbasis *bagging ensemble* yang popular. terdiri atas banyak pohon keputusan (*decission tree*). *Random forest.*

merupakan hasil voting dari masing-masing tree. Seandainya setiap classifier dalam ensemble merupakan *decision tree classifier*, maka kumpulan *classifier* adalah sebuah "*forest*" [9].

Langkah-langkah rinci dari *Random Forest* adalah sebagai berikut :

1. Menghasilkan Sebuah set-set pelatihan baru dengan sampel acak dengan penggantian (*bootstrap*) dari set pelatihan asli.
2. Untuk setiap set pelatihan baru, dibangun sebuah *tree* dengan pemilihan fitur acak di setiap simpul *tree* dan tanpa pemangkasan.
3. Setelah sejumlah besar *tree* dihasilkan, data baru diprediksi dengan menggabungkan hasil semua *tree*, dengan strategi voting mayoritas.

2.6. Optuna

Optuna merumuskan optimasi *hyperparameter* sebagai proses meminimalkan atau memaksimalkan fungsi objektif yang mengambil sekumpulan hyperparameter sebagai input dan mengembalikan skor validasinya. Optuna mulai menggabungkan parameter yang berbeda dan menguji algoritma untuk melihat apakah kombinasi parameter mengarah pada peningkatan algoritma. Di akhir proses pengujian parameter, model *machine learning* yang optimal kembali, yaitu model dengan metrik terbaik [18]. Prinsip *define-by-run* memungkinkan pengguna untuk secara dinamis membangun ruang pencarian dengan cara yang tidak pernah mungkin dilakukan dengan kerangka kerja penyetelan *hyperparameter* sebelumnya. Kombinasi algoritma pencarian dan pemangkasan yang efisien sangat meningkatkan efektivitas optimasi. Desain yang dapat diskalakan dan serbaguna memungkinkan pengguna dari berbagai jenis untuk menerapkan kerangka kerja untuk berbagai tujuan [19].

2.7. *Confusion Matrix*

*Confusion Matrix* digunakan untuk mengevaluasi atau mengukur kinerja sebuah model dengan *Accuracy, Precision, Recall*, dan *F-Score* [20], sebagai berikut.

1. *Accuracy* mengukur proporsi jumlah total klasifikasi yang benar.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

2. *Precision* mengukur jumlah klasifikasi yang benar yang dihukum oleh jumlah klasifikasi yang salah.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (7)$$

3. *Recall* mengukur jumlah klasifikasi yang benar yang dihukum oleh jumlah entri yang tidak terjawab.

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

4. *F-score* mengukur rata-rata harmonik presisi dan daya ingat, yang berfungsi sebagai derived. pengukuran efektivitas.

$$F\text{-}score = \frac{FP}{FP + TN} \quad (9)$$

## 3. HASIL DAN PEMBAHASAN

Mengikuti metodologi pada gambar 1, dalam penelitian ini pertama, peneliti menggabungkan set pelatihan dengan jumlah 175.341 *record*, dan set pengujian dengan jumlah 82.332 record, dan total data set adalah 257.673 record. Kemudian melakukan penanganan pada *imbalance class* yaitu *binary class* dengan menggunakan teknik ADASYN lalu melakukan *split data*. Selanjutnya melakan *hyperparameter tuning* menggunakan optuna dengan *parameter n_estimator* (580, 600, 10), *criterion (gini*, *entropy*), *max_features (auto, sqrt), max_depth* (2, 50). Dan menerapkan nya pada algoritma *random forest*.

3.1. Perbandingan Split Data

Peneliti melakukan penelitian dengan menggunakan beberapa macam *split data* sebagai berikut :

Tabel 2 Perbandingan *split* data

| ***Split*** | ***Accuracy*** | ***Precision*** | ***Recall*** | ***F1-score*** |
|---|---|---|---|---|
| 60/40 | 99,75 | 99,88 | 99,62 | 99,75 |

| | | | | |
|---|---|---|---|---|
| 70/30 | 99,76 | 99,90 | 99,62 | 99,76 |
| 80/20 | 99,80 | 99,90 | 99,70 | 99,80 |
| 90/10 | 99,86 | 99,93 | 99,78 | 99,86 |

Pada tabel 2 dapat dilihat *accuracy* tertinggi didapat pada *split* data 90/10.

3.2. Perbandingan Metode *Imbalance Data*

Selain melakukan penelitian dengan menggunakan beberapa macam *split*, peneliti juga melakukan perbandingan metode *imbalance data* sebagai berikut :

Tabel 3 Perbandingan metode *imbalance* Mengggunakan *Random Forest* pada *dataset* UNSW-NB15

| Method | Accuracy |
|---|---|
| *SMOTE* [21] | 95,1 % |
| *Clustering under-sampling* [22] | 97,1 % |
| *Random Over-sampling* [23] | 87,82 % |
| *Random Forest [14]* | 98,86 % |

Dapat dilihat pada tabel 3 metode yang diusulkan mendapat kan akurasi yang lebih baik dari pada metode yang lainnya.

## 4. KESIMPULAN

Berdasarkan hasil pengujian dengan teknik ADASYN untuk penanganan *Imbalance data* pada *Binarry Class* dan menggunakan model algoritma *Random Forest*, serta *Hyperparameter Optuna* untuk klasifikasi *Anomali* pada data UNSW-NB15 membuktikan adanya peningkatan sebelum dan sesudah menggunakan metode yang diusulkan, ini menjadi pengamatan yang efektif untuk pemodelan prediksi. Dari data UNSW-NB15 dengan menggunakan metode yang diusulkan dengan pembagian data 60/40, 70/30, 80/20, dan 90/10 mendapatkan nilai akurasi tertinggi pada split data 90/10 dengan hasil 99.86%.

## DAFTAR PUSTAKA


[1] F. Alkhudhayr and S. Elkhdiri, "Information Security : A review of information security issues and techniques," *2019 2nd Int. Conf. Comput. Appl. Inf. Secur.*, pp. 1–6, 2019.

[2] J. Jinquan, M. A. Al-Absi, A. A. Al-Absi, and H. J. Lee, "Analysis and Protection of Computer Network Security Issues," *Int. Conf. Adv. Commun. Technol. ICACT*, vol. 2020, pp. 577–580, 2020, doi: 10.23919/ICACT48636.2020.9061266.

[3] B. Prasetyo, V. Suryani, and D. R. Anbiya, "Analisis Deteksi Malware pada Aplikasi Android Fintech berdasarkan Permissions dengan menggunakan Naive Bayes dan Random Forest," vol. 8, no. 5, pp. 9885–9897, 2021.

[4] Z. Hisyam, U. A. Yogyakarta, C. Catur, and A. M. Yogyakarta, "Implementasi Network Intrusion Detection System (NIDS) Dalam Sistem Keamanan Open Cloud Computing," vol. 17, no. 2, pp. 1–9, 2019.

[5] O. Ibitoye, R. Abou-Khamis, A. Matrawy, and M. Omair Shafiq, "The threat of adversarial attacks on machine learning in network security - A survey," *arXiv*, 2019.

[6] J. Li *et al.*, "SMOTE-NaN-DE: Addressing the noisy and borderline examples problem in imbalanced classification by natural neighbors and differential evolution," *Knowledge-Based Syst.*, vol. 223, 2021, doi: 10.1016/j.knosys.2021.107056.

[7] L. Pan and X. Xie, "Network Intrusion Detection Model Based on PCA + ADASYN and XGBoost," *ACM Int. Conf. Proceeding Ser.*, pp. 1–5, 2020, doi: 10.1145/3453187.3453311.

[8] J. Liu, Y. Gao, and F. Hu, "A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM," *Comput. Secur.*, vol. 106, p. 102289, Jul. 2021, doi: 10.1016/j.cose.2021.102289.

[9] S. D. B. Kurnia budi, "Seleksi Fitur dengan Information Gain untuk Meningkatkan Deteksi Serangan DDoS Menggunakan Random Forest," vol. 19, no. 1, pp. 56–66, 2020.

[10] B. U. of technology Chao Lu, "Telecom Fraud Identification Based on ADASYN and Random Forest," pp. 447–452, 2020.

[11] H. A. Ahmed, A. Hameed, and N. Z. Bawany, "Network intrusion detection using oversampling technique and machine learning algorithms," *PeerJ Comput. Sci.*, vol. 8, 2022, doi: 10.7717/PEERJ-CS.820.

[12] M. N. Aziz and T. Ahmad, "Clustering under-sampling data for improving the performance of intrusion detection system," *J. Eng. Sci. Technol.*, vol. 16, no. 2, pp. 1342–1355, 2021.

[13] M. Azizjon, A. Jumabek, and W. Kim, "1D CNN based network intrusion detection with normalization on imbalanced data," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2020, pp. 218–224, doi: 10.1109/ICAIIC48513.2020.9064976.

[14] O. Faker and E. Dogdu, "Intrusion Detection Using Big Data and Deep Learning Techniques," in *Proceedings of the 2019 ACM Southeast Conference*, 2019, pp. 86–93, doi: 10.1145/3299815.3314439.

[15] N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems : Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set The evaluation of Network Anomaly Detection Systems : Statistical analysis of," vol. 3555, no. January, pp. 0–14, 2016, doi: 10.1080/19393555.2015.1125974.

[16] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *2015 Mil. Commun. Inf. Syst. Conf. MilCIS 2015 - Proc.*, 2015, doi: 10.1109/MilCIS.2015.7348942.

[17] T. Xu, G. Coco, and M. Neale, "A predictive model of recreational water quality based on adaptive synthetic sampling algorithms and machine learning," *Water Res.*, vol. 177, p. 115788, 2020, doi: 10.1016/j.watres.2020.115788.

[18] I. Vaccari, S. Narteni, M. Aiello, M. Mongelli, and E. Cambiaso, "Exploiting Internet of Things Protocols for Malicious Data Exfiltration Activities," *IEEE Access*, vol. 9, pp. 104261–104280, 2021, doi: 10.1109/ACCESS.2021.3099642.

[19] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 2623–2631, 2019, doi: 10.1145/3292500.3330701.

[20] B. Hu, J. Wang, Y. Zhu, and T. Yang, "Dynamic deep forest: An ensemble classification method for network intrusion detection," *Electron.*, vol. 8, no. 9, 2019, doi: 10.3390/electronics8090968.