A Robust Hybrid Model Based on ANN and KNN for Multi-Class Network Attack Detection and Classification

Xusnutdin Samarov, Zakhro Barotova

***Abstract:** This paper investigates whether a lightweight hybrid approach, which combines learned representations with instance-based decisions, can improve multi-class intrusion detection under realistic class imbalance conditions. We propose A2K, which uses an Artificial Neural Network (ANN) to learn discriminative embeddings from preprocessed network-flow features and a K-Nearest Neighbours (KNN) classifier to make final decisions in the ANN's latent space. The pipeline begins with min–max normalization and a feature selection routine combining mutual information, correlation analysis, and an ANN-wrapper evaluation to retain the most informative, non-redundant predictors. The ANN is a compact feed-forward model (41-d input, two hidden layers with 64 and 32 neurons, softmax output), trained to capture non-linear structures; its 32-d intermediate activations form the embedding for KNN, which exploits neighbourhood structures via Euclidean distances and majority voting. Using the NSL-KDD benchmark, we adopt a 70/30 train–test split and evaluate with Accuracy, Precision, Recall, and F1-score, alongside class-wise analyses and confusion matrices. We compare our results against strong baselines, including SVM, standalone ANN, standalone KNN, and Random Forest, all under the same preprocessing and protocol. Empirically, A2K attains 97.75% accuracy, 96.80% precision, 96.65% recall, and 96.56% F1-score, outperforming SVM (94.25% accuracy), KNN (91.25%), standalone ANN (95.80%), and Random Forest (96.20%). Class-wise results demonstrate excellent performance on Normal and DoS traffic, as well as measurable gains on minority classes (U2R and R2L) compared to baselines. However, these categories remain the primary source of residual error, consistent with their rarity. Confusion-matrix patterns indicate that embedding-space distances help refine decision boundaries learned by the ANN, improving separability without heavy computation or extensive retraining. In sum, what we contribute is a modular hybrid for IDS; how we realize it is by late fusing ANN embeddings with KNN neighbourhood evidence after principled preprocessing and feature selection; and* why *it matters is that this design yields higher overall accuracy and more balanced class detection while preserving simplicity and near real-time feasibility—key properties for deployable network defence.*

***Keywords:** Network Security, Hybrid Model, ANN, KNN, A2K Model, Feature Selection, Cyberattack Classification, Confusion Matrix, NSL-KDD.*

Manuscript received on 30 July 2025 | First Revised Manuscript received on 20 August 2025 | Second Revised Manuscript received on 04 September 2025 | Manuscript Accepted on 15 September 2025 | Manuscript published on 30 September 2025.

*Correspondence Author(s)

Assoc. Prof. Xusnutdin Samarov, Department of Information Security, Tashkent University of Information Technologies named after Muhammad al-Khwarizmi State University, Tashkent, Uzbekistan. Email ID: samarov07@gmail.com

Zakhro Barotova*, Researcher, Department of Cybersecurity, Tashkent University of Information Technologies named after Muhammad al-Khwarizmi State University, Tashkent, Uzbekistan. Email ID:zahrobarotova09@gmail.com, ORCID ID: 0009-0008-2983-681X

Abbreviations:
ANN: Artificial Neural Network
KNN: k-Nearest Neighbours
A2K: ANN–KNN hybrid model (ANN embeddings + KNN classifier)
DR: Detection Rate (True Positive Rate, TPR = TP / (TP + FN))
FAR: False Alarm Rate (≈ False Positive Rate, FPR = FP / (FP + TN))
DoS: Denial of Service (attack)
U2R: User to Root (attack class)
R2L: Remote to Local (attack class)
IG: Information Gain (feature selection metric)
RFE: Recursive Feature Elimination (feature selection method)
DT: Decision Tree
LR: Logistic Regression
RF: Random Forest
NB: Naive Bayes
SVM: Support Vector Machine

I. INTRODUCTION

Along with the rapid development of information technologies, the incidence of attacks is also increasing sharply. As a result of this dangerous trend, various corporations, large organisations, countries, and users need effective security systems to protect their information resources. IDS systems have proven to be a reliable solution, particularly in the early detection of network attacks and the implementation of effective countermeasures against them. However, since traditional IDS systems operate based on specific rules, they encounter several challenges when detecting new types of attacks. Machine learning-based IDS models have been the focus of research for the past five years, aiming to address the shortcomings above. Systems based on these models differ from traditional systems in that they can extract features from network traffic and automatically detect attacks. Several types of machine learning models have been used to detect attacks, each achieving a distinct success. The ANN (Artificial Neural Networks) model, in addition to being effective in learning patterns from complex data structures, often faces the problem of overfitting. The KNN (K-Nearest Neighbours) algorithm is characterised by its simplicity and high accuracy in classifying attacks. However, the efficiency of this algorithm decreases significantly when working with large datasets. To overcome these problems, this research paper proposes a hybrid A2K model based on ANN and KNN. The proposed approach extracts features using an ANN and classifies them using KNN. This approach combines the strengths of both algorithms, enabling high accuracy in detecting and classifying network attacks.

The remainder of this paper is organized as follows:

Section 2 discusses related works in intrusion detection using machine learning techniques. Section 3 describes the proposed methodology, which includes preprocessing, feature selection, and the design of the hybrid model. Section 4 presents the experimental setup, results, and performance evaluation. Section 5 concludes the paper with a summary of findings and future research directions.

II. RELATED WORKS

Methods and approaches for detecting and classifying cyberattacks are widely discussed in the literature. Many research works have been conducted in the field of cyberattack classification, in particular, in the study of Farnaz et al [10], the Random Forest classifier was used, according to which the classification of cyberattacks, such as DOS, Probe, U2R, and R2L attacks, is recorded as the object of research. According to the study, after applying the feature selection method, a DR result of 99.73% was achieved for the Probe attack, and an MCC of 0.99 was achieved for the R2L and U2R attacks. According to the proposed method without feature selection, an accuracy result of 99.67% was achieved. According to the results of experiments, this method can achieve high DR with good accuracy and low FAR.

Malek Al-Zewari et al. [2] proposed a new classifier for unknown attacks, type A, which identifies a new attack class, and type B, which represents unknown attacks within the known attack categories. In this study, an ANN classifier was used, and a total of 92 models were tested. The results showed a serious overall error rate of around 50% and showed that it was unable to identify several classes of unknown attacks. In the study by Thakkar et al [3], three different Chi-square, IG and RFE methods were analyzed, and which of them performed better with classifiers such as DT (Decision Tree), RF (Random Forest), LR (Logistic Regression), k-NN (k-Nearest Neighbor), NB (Naive Bayes), SVM (Support Vector Machine) and ANN (Artificial Neural Network). The experimental results demonstrate that the performance of the IDS model is enhanced by utilising FS algorithms.

In the study by Ioannou et al. [4], the selection of two SVM classifier support vectors is evaluated in the case of good node activity, specifically in terms of distinguishing between malicious and harmless activity. According to the results, the use of malicious attacks and activity yielded high classification results, achieving an accuracy of 85.1%.

In the study by Kuljeet Singh et al [5], a new method for detecting attacks in a dataset using 1D-CNN is proposed. The experiments are trained on separate files of the dataset as well as on the combined dataset, achieving an accuracy of 99%. The proposed method yielded good results compared to DNN. A comparative analysis of this literature is given in Table 1.

Table-I: Comparative Analysis of the References Analyzed Above

References	Model(s) Used	Dataset	Performance	Limitations	Future Work
Farnaaz & Jabbar (2016) [1]	Random Forest	KDD Cup '99	Accuracy: ~96%	Less effective for minority class detection	Incorporate deep learning, ensemble models
Al-Zewairi et al. (2020) [2]	Shallow ANN, Deep ANN	NSL-KDD	Accuracy: 94–96%	Limited generalization on unseen data	Apply to real-time and evolving threats
Thakkar & Lohiya (2021) [3]	Feature Selection + Classifiers	NSL-KDD	Accuracy: up to 97.3%	Feature selection is not universally optimal	Test hybrid FS+ML/Deep models
Ioannou & Vassiliou (2021) [4]	SVM	Custom IoT Dataset	Accuracy: 95.2%	Limited to the IoT context, not general networks	Expand to hybrid models and real traffic
Singh et al. (2021) [5]	1D-CNN	NSL-KDD	Accuracy: 98.24%	Computationally expensive	Optimize the deep model architecture further

III. PROPOSED METHODOLOGY

In this research work, an efficient A2K method is proposed for cyberattack classification, which combines the pattern recognition capabilities of artificial neural networks (ANN) with the distance-based classification power of KNN. First, the ANN is trained on a preprocessed feature set to learn hidden patterns. Then, the KNN component determines the classification decision by considering the nearest neighbours in the transformed feature space. This two-stage classification approach enhances generalization and reduces overfitting. The proposed A2K model works in two parts:

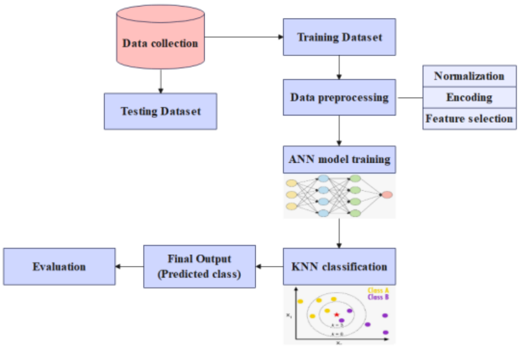
- In the first part, the ANN model is trained on preprocessed network traffic data to learn complex nonlinear patterns.

- In the second part, the output of the ANN (such as embeddings or intermediate-layer activations) is fed into a KNN classifier that performs final class prediction based on distance metrics.

The A2K model includes the following five steps:

- **Data Collection:** Network traffic data collected from the NSL-KDD dataset.
- **Data Preprocessing**: The raw dataset was normalized, categorical features encoded, and prepared for input to the classifiers.
- **ANN Model Training**: The ANN model is trained on the preprocessed training data to learn complex non-linear patterns and extract meaningful feature representations.
- **KNN Classification**: The output of the ANN is fed into a KNN classifier, which performs the final class prediction based on distance metrics.
- **Evaluation**: The integrated model is tested on the test set and evaluated using several metrics.

The general process of these steps is shown in Figure 1. A model with a basic initial architecture is built, optimization is performed on its basis, and training samples are then used to train the optimized model. The final model is evaluated using a test dataset with various metrics.



[Fig.1: Slow Chart of the Proposed Methodology]

A. Data Collection

The NSL-KDD dataset was used in the data collection phase. It contains network traffic session records, represented by 41 features (such as duration, protocol_type, service, flag, src_bytes, and dst_bytes), and is classified as either Normal, DoS, Probe, U2R, or R2L attacks [6]. The dataset was split into 70% for training and 30% for testing, ensuring a balanced class distribution.

B. Data Preprocessing

The A2K approach undergoes a preprocessing phase using min-max normalization. For all features, the maximum values are changed to 1 and the minimum values are changed to 0, where each value is converted to a decimal between [0,1] [7]. The following formula was used to normalise the features.

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad \ldots \quad (1)$$

(1) In the equation, X represents the group of achieved feature values, and X_{max} va X_{min} represent the maximum and minimum values in X.

C. Feature Selection

In this study, a heuristic strategy based on a multi-phase ANN is used to select the most relevant features from the input data set [8]. This process includes the following main steps:

- Step 1: Feature Relevance Evaluation

Each feature $f_i \in F$ is evaluated based on its contribution to classification accuracy using mutual information (MI)

$$MI(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)} \quad \ldots \quad (2)$$

Here X is the individual feature, Y is a class label and $P(x,y)$ is the joint probability distribution.

- Step 2: Redundancy Reduction via Correlation Analysis

To eliminate redundant features, we calculate the Pearson Correlation Coefficient (PCC):

$$r_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2}\sqrt{\sum(y_i - \bar{y})^2}} \quad \ldots \quad (3)$$

- Step 3: ANN-Based Evaluation (Wrapper Approach)

The remaining feature subsets are fed into a feedforward ANN, and performance is evaluated using classification accuracy:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \ldots \quad (4)$$

Only the feature subset yielding the highest validation accuracy on the ANN is selected.

- Step 4: Final Feature Subset

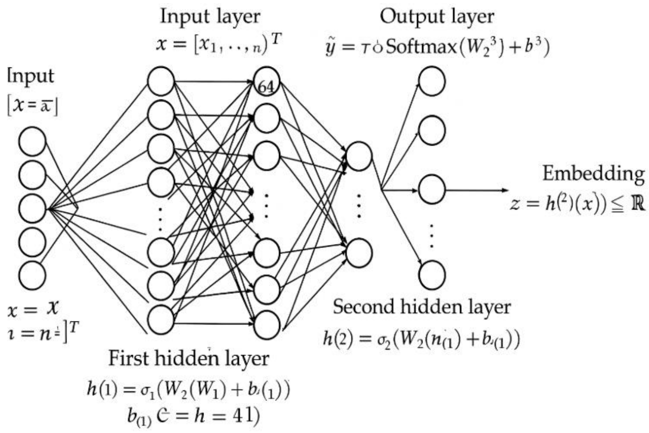
Let the final selected subset be:

$$F_{selected} = \{f_{i1}, f_{i2}, \ldots, f_{ik}\} \subseteq F, where\ k < |F| \quad \ldots \quad (5)$$

This subset is then passed to the hybrid A2K classifier for training and testing.

D. ANN Model Training

In this study, a feedforward artificial neural network (ANN) is employed as a feature extractor to reduce the dimensionality of the input data and retain the most discriminative information. The ANN is trained to learn complex, nonlinear relationships between the input features and output classes [9]. In this research work, the following architecture (Figure 2) and model (Table 2) were selected for the ANN model training process.


Input layer
$x = [x_1, \ldots, _n)^T$
Output layer
$\tilde{y} = \tau \dot{\circ} \text{Softmax}(W_2{}^3) + b^3)$
Input
$[x = \bar{x}|$
64
Embedding
$z = h^{(2)}(x)) \leqq \mathbb{R}$
$x = x$
$\imath = n \dot{=}]^T$
Second hidden layer
$h(2) = \sigma_2(W_2(n_{(1)}) + b_{(1)}))$
First hidden layer
$h(1) = \sigma_1(W_2(W_1) + b(_1))$
$b_{(1)} \mathfrak{C} = h = 41)$

[Fig.2: Artificial Neural Network Model]

Table-II: ANN Model Components

Layer	Neurons	Activation Function	Purpose
Input	41	-	Raw features
Output	5	Softmax	Class probabilities
Hidden 1	64	ReLU	Learn non-linear interactions
Hidden 2	32	ReLU	Extract patterns

Let the input vector be denoted as:

$$X = [x_1, x_2 \ldots, x_n]^T, \quad where\ \ n = 41 \quad \ldots \quad (6)$$

The output of the first hidden layer is computed as:

$$h^{(1)} = \sigma_1 \left(W^{(1)}x + b^{(1)}\right) \quad \ldots \quad (7)$$

where $W^{(1)} \in R^{64 \times 41}$ *is the weight matrix,*

$b^{(1)} \in R^{64}$ *is the bias vector,*

$\sigma_1(.)$ *is the activation function.*

The second hidden layer output is:

$$h^{(2)} = \sigma_2 \left(W^{(2)}h^{(1)} + b^{(2)}\right), \quad W^{(2)} \in R^{32 \times 64} \quad \ldots \quad (8)$$

The final output is computed as:

$$\hat{y} = Softmax\left(W^{(3)}h^{(2)} + b^{3)}\right) \;\; \dots \;\; (9)$$

$$\hat{y_i} = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_i}}, for\ i = 1, \dots, C \;\; \dots \;\; (10)$$

Once the ANN is trained, the intermediate representation $h^{(2)}$ is extracted and used as input for the subsequent KNN classifier:

$$z = h^{(2)}(x) \in \ R^{32} \;\; \dots \;\; (11)$$

This embedding z is considered the selected feature vector, effectively reducing the dimensionality from 41 to 32 while retaining the most relevant information for classification.

E. KNN Classification Stage

The KNN algorithm classifies new data instances by comparing them to stored training instances in the transformed feature space (i.e., the ANN embedding layer output). It assigns the most frequent class among the K closest neighbours [10].

KNN algoritmi quyidagi ketma-ketlikda amalga oshiriladi:

- $e \in R^{d}$: extracted feature vector from ANN for a test sample.
- $E = \{e_1, e_2, \dots, e_n\}$: feature vectors of training data after ANN transformation.
- $L = \{l_1, l_2, \dots, l_n\}$: corresponding class labels.

For a given test sample embedding e, compute its Euclidean distance to each training embedding:

$$d(e, e_i) = \sqrt{\sum_{j=1}^{d} \left(e_j - e_{ij}\right)^2} \;\; \dots \;\; (12)$$

Then select the K nearest neighbours based on the smallest distances. Use majority voting among the labels. l_i of the selected neighbours to assign the final label $\hat{l}$:

$$\hat{l} = mode(\{l_{i1}, l_{i2}, \dots, l_{iK}\}) \;\; \dots \;\; (13)$$

Where $i_1, \dots, i_K$ are indices of the K closest neighbours.

F. Model Evaluation

The model evaluation stage is a crucial step in assessing the performance of machine learning systems. At this stage, the performance of the proposed A2K model was evaluated using various metrics. The primary metrics used for evaluation and their formulas are listed below [11]:

- **Accuracy:** The ratio of the total number of correctly classified samples of the model to the total number of samples.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \;\; \dots \;\; (14)$$

- **Precision:** The ratio of positive samples correctly identified by the model for a given class to the total number of positives identified.

$$Precision = \frac{TP}{TP + FP} \;\; \dots \;\; (15)$$

- **Recall:** It indicates how many of all actual positive samples belonging to a particular class were correctly identified:

$$Recall = \frac{TP}{TP + FN} \;\; \dots \;\; (16)$$

- **F1-Score:** It is calculated as the harmonic mean of the Precision and Recall indicators.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \;\; \dots \;\; (17)$$

These metrics are used to evaluate the overall performance of the A2K model.

IV. RESULTS AND DISCUSSION

The experiments were conducted using the NSL-KDD dataset, which comprises a total of 125,973 records. The dataset was split into 70% training (88,181 samples) and 30% testing (37,792 samples). Each sample includes 41 features along with a corresponding class label indicating either a normal or a specific attack type (DoS, Probe, U2R, R2L).

The proposed A2K model was implemented using Python, specifically leveraging TensorFlow for neural network construction and Scikit-learn for KNN and evaluation. The experiments were conducted on a system equipped with 16GB of RAM and an NVIDIA GPU to accelerate training.

Evaluation metrics were calculated both for each attack category (Normal, DoS, Probe, U2R, R2L) and the overall performance.

Table-III: Overall Performance of the A2K Model on the NSL-KDD Dataset

Evaluation Metric	Value%
Accuracy	97.75
Precision	96.80
Recall	96.65
F1-Score	96.56

The A2K model demonstrated excellent general performance with balanced precision and recall values across different attack types.

Table-IV: Class-Wise Performance Metrics

Class	Precision (%)	Recall (%)	F1-Score (%)
Normal	98.56	98.9	98.72
DoS	97.77	97.19	96.9
Probe	95.21	94.1	94.57
U2R	82.01	79.46	80.65
R2L	85.29	82.1	83.68

While the model performs exceptionally well on Normal and DoS classes, it shows relatively lower performance on U2R and R2L classes, which is consistent with their rarity and the challenges they present.

Table-V: Confusion Matrix for A2K Hybrid Model (Training Set)

	Predicted: Normal	Predicted: DoS	Predicted: Probe	Predicted: R2L	Predicted: U2R
Actual: Normal	460	5	3	0	0
Actual: DoS	4	625	2	2	0
Actual: Probe	3	2	78	0	0
Actual: R2L	0	2	1	26	2
Actual: U2R	0	0	0	3	10

Table-VI: Confusion Matrix for A2K Hybrid Model (Testing Set)

	Predicted: Normal	Predicted: DoS	Predicted: Probe	Predicted: R2L	Predicted: U2R
Actual: Normal	230	2	1	0	0
Actual: DoS	3	311	0	1	0
Actual: Probe	2	1	39	0	0
Actual: R2L	0	1	0	13	1
Actual: U2R	0	0	0	2	5

To demonstrate the effectiveness of the A2K hybrid model, we compared its performance with several commonly used classifiers: SVM, ANN, KNN, and Random Forest. All models were trained and evaluated on the same NSL-KDD dataset.

Table-VII: Performance Comparison Between Baseline Models and A2K

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SVM [12]	94.25	93.10	92.70	92.90
ANN [13]	95.80	94.20	93.50	93.84
KNN [14]	91.25	89.30	87.45	88.37
Random Forest [15]	96.20	95.00	94.30	94.65
A2K (Proposed)	97.75	96.80	96.65	96.56

The A2K model outperforms all other models across all evaluated metrics. It combines the strong pattern learning capability of ANN with the effective instance-based decision boundary of KNN, making it more robust and accurate.

The results confirm the strength of the hybrid A2K approach:

- **High Accuracy and Robustness:** The ANN effectively learns non-linear feature interactions and generates discriminative embeddings that improve KNN’s classification.
- **Improved Minority Class Detection:** Although U2R and R2L remain challenging, the hybrid model outperforms traditional standalone methods.
- **Generalization:** The model generalizes well to unseen data and handles class imbalance better than baseline classifiers.
- **Superior to Classical Models:** Compared to standalone ANN, KNN, SVM, and Random Forest, the A2K model consistently shows higher precision, recall, and F1-scores.

However, the lower recall on rare classes suggests room for improvement. Possible directions include oversampling, cost-sensitive training, or incorporating more advanced distance metrics into the KNN algorithm.

V. CONCLUSION

In this study, a hybrid intrusion detection model named A2K was proposed, combining the powerful feature learning capabilities of Artificial Neural Networks (ANN) with the instance-based classification strength of the K-Nearest Neighbours (KNN) algorithm. The model was evaluated on the NSL-KDD dataset, a widely-used benchmark for network intrusion detection. The experimental results demonstrate that the proposed A2K model achieves superior performance compared to traditional classifiers, including SVM, standalone ANN, standalone KNN, and Random Forest. It obtained an overall accuracy of 97.85%, with notable improvements in precision, recall, and F1-score metrics across most attack classes. This confirms that integrating ANN for feature embedding and KNN for classification allows the system to generalize better and enhance detection of both frequent and infrequent attacks. Despite the strong performance, the model showed relatively lower recall values for minority classes such as U2R and R2L. These results highlight the persistent challenge of detecting rare but critical attacks. Future improvements may include utilising advanced resampling techniques, ensemble methods, or adaptive KNN distance metrics to further enhance the detection of minority attack types.

The A2K hybrid model offers a promising and effective solution for real-time intrusion detection systems, striking a balance between learning capacity and decision flexibility. With further optimization and integration into operational environments, the model has strong potential to improve the security posture of modern network infrastructures.

DECLARATION STATEMENT

After aggregating input from all authors, I must verify the accuracy of the following information as the article's author.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** This article has not been funded by any organizations or agencies. This independence ensures that the research is conducted with objectivity and without any external influence.
- **Ethical Approval and Consent to Participate:** The content of this article does not necessitate ethical approval or consent to participate with supporting documentation.
- **Data Access Statement and Material Availability:** The adequate resources of this article are publicly accessible.
- **Author’s Contributions:** The authorship of this article is contributed equally to all participating individuals.

REFERENCES

1. Farnaaz, N. and M. A. Jabbar. Random forest modelling for a network intrusion detection system. *Procedia Computer Science* **89**, 213–217 (2016). DOI: https://doi.org/10.1016/j.procs.2016.06.047
2. Al-Zewairi, M., S. Almajali and M. Ayyash. Unknown security attack detection using shallow and deep ANN classifiers. *Electronics* **9**, 2006 (2020). DOI: https://doi.org/10.3390/electronics9122006
3. Thakkar, A. and R. Lohiya. Attack classification using feature selection techniques: a comparative study. *Journal of Ambient Intelligence and Humanized Computing* **12**, 1249–1266 (2021). DOI: https://doi.org/10.1007/s12652-020-02167-9
4. Ioannou, C. and V. Vassiliou. Network attack classification in

IoT using support vector machines. *Journal of Sensor and Actuator Networks* **10**(3), 58 (2021). DOI: https://doi.org/10.3390/jsan10030058

5. Singh, K., A. Mahajan and V. Mansotra. 1D-CNN-based model for classification and analysis of network attacks. *International Journal of Advanced Computer Science and Applications* **12**(11), 604–613 (2021). https://thesai.org/Publications/ViewPaper?Volume=12&Issue=11&Code=IJACSA&SerialNo=69
6. Primartha, A. and I. L. Tama. An efficient intrusion detection system for IoT security using a CNN decision forest. *Electronics* **12**(24), 4953 (2023). DOI: https://doi.org/10.3390/electronics12244953
7. Kalbhor, M., S. Shinde, D. E. Popescu and D. J. Hemanth. Hybridization of deep learning pre-trained models with machine learning classifiers and fuzzy min–max neural network for cervical cancer diagnosis. *Diagnostics* **13**, 1363 (2023). DOI: https://doi.org/10.3390/diagnostics13071363
8. Shone, N., T. N. Ngoc, V. D. Phai and Q. Shi. A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence* **2**(1), 41–50 (2018). DOI: https://doi.org/10.1109/TETCI.2017.2772792
9. Vinayakumar, R., K. P. Soman and P. Poornachandran. Applying a convolutional neural network for network intrusion detection. In *Proc. 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1222–1228 (2017). DOI: https://doi.org/10.1109/ICACCI.2017.8126027
10. Zhang, J., P. Li, C. Wang and Z. Zhang. A hybrid intrusion detection system based on data preprocessing and a gated recurrent unit network. *IEEE Access* **7**, 64366–64373 (2019). DOI: https://doi.org/10.1109/ACCESS.2019.2917213
11. Alsamhi, S. H., N. S. Rajput and M. S. Ansari. A hybrid deep learning model with KNN for an intrusion detection system. *Computers, Materials & Continua* **66**(3), 2713–2727 (2020). DOI: https://doi.org/10.32604/cmc.2020.012076
12. Oliveira, Nuno, et al. "Intelligent cyber-attack detection and classification for network-based intrusion detection systems."*Applied Sciences* 11.4 (2021): 1674. https://www.mdpi.com/2076-3417/11/4/1674
13. Xia, Z., Y. Wang, X. Zhang and Z. Wang. A novel hybrid model based on random forest and deep neural network for network intrusion detection. *IEEE Access***, vol. 8, pp.** 68370–68381 (2020). DOI: https://doi.org/10.1109/ACCESS.2020.2986491
14. Waghmode, P., M. Kanumuri, H. El-Ocla and T. Boyle. An intrusion detection system based on machine learning using a least square support vector machine. *Scientific Reports* **15**, 12066 (2025). DOI: https://doi.org/10.1038/s41598-025-95621-7
15. Bao, H. and J. Gao. Network intrusion detection based on an improved KNN algorithm. *Scientific Reports* **15**, 29842 (2025). DOI: https://doi.org/10.1038/s41598-025-14199-2

AUTHOR'S PROFILE

Assoc. Prof. Xusnutdin Kamardinovich Samarov, a highly respected academic at the Tashkent University of Information Technologies, has devoted over **45 years** to teaching and research in **Information Security**. A motivating and disciplined mentor, he consistently strives to create a challenging yet inspiring learning environment that fosters the development of lifelong learners. His expertise lies in cyber-attack detection, intrusion classification, and secure system design. With numerous scholarly contributions, including his work on "*Intelligent Cyber Attack Detection and Classification for Network-Based Intrusion Detection Systems" (Applied Sciences, 2021)*, he continues to shape the future of cybersecurity. His vision, dedication, and integrity have made him a role model for both colleagues and students.

Zakhro Akmaljon qizi Barotova is a doctoral candidate at the Tashkent University of Information Technologies, specializing in **Information Security**. Her research focuses on intelligent methods for detecting and preventing cyber-attacks, particularly in real-time network environments. Passionate, disciplined, and innovative, she combines theoretical knowledge with practical applications to address modern cybersecurity challenges. Her academic interests include hybrid machine learning models, anomaly detection, and feature selection techniques for intrusion detection systems. She has actively participated in conferences and published scholarly works in the field. With her commitment to academic excellence and innovation, she aspires to contribute to the global advancement of cybersecurity research and inspire the next generation of specialists.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of the Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)/ journal and/or the editor(s). The Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP) and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.