






## Research Article

## An optimized model for network intrusion detection in the network operating system environment

Abbas A. Abdulhameed<sup>1</sup>, , Sundos A. Hameed Alazawi<sup>1,\*</sup>, , Ghassan Muslim Hassan<sup>2</sup>, <sup>1</sup> Department of Computer Science, Mustansiriyah University, Baghdad, 10052, Iraq<sup>2</sup> Ashur University, Baghdad, Iraq

## ARTICLE INFO

## Article History

Received 17 Jul 2024

Accepted 15 Oct 2024

Published 07 Nov 2024

## Keywords

Cybersecurity

Feature Selection

Intrusion Detection

Network Security

Optimisation



## ABSTRACT

With the heavy reliance on computers and information technology to send and receive data across networks of various types, there has been concern about securing that data from intrusions and cyber-attacks. The expansion of network usage has led to an increase in hacker attacks, which has led to prioritizing cybersecurity precautions in detecting potential threats. Intrusion detection techniques are a critical security measure to protect networks in both personal and corporate environments that are managed by network operating systems. For this, the paper relies on designing a network intrusion detection model. Since deep neural networks (DNNs) are classic deep learning models known for their strong classification performance, making them popular in intrusion detection along with other machine learning algorithms, they have been chosen to improve intrusion classification models based on datasets for intrusion detection systems. The basic structure of this proposal is to adopt one of the optimization algorithms in extracting features from the dataset to obtain more accurate results in the classification and intrusion detection stage. The developed Corona Virus algorithm is adopted to improve the system performance by identifying optimal features. This algorithm, which consists of several stages, optimally selects individuals based on features from the NSL-KDD dataset used for intrusion detection. The resulting optimization solution acts as a network structure for the intrusion classification model based on machine learning and deep learning algorithms. The test results showed exceptional performance on the NSL-KDD dataset, where the proposed Convolution Neural Network CNN model achieved 99.3% accuracy for multi-class classification, while the Decision Tree (DT) achieved 88.64% accuracy for anomaly detection in bi-class classification.

## 1. INTRODUCTION

Computers have revolutionised our lives and work, becoming ubiquitous through various devices, from desktops to smartphones. This widespread adoption has raised significant concerns about securing the vast amounts of data exchanged over networks [1]. Most network attacks involve fundamental issues, such as hacking the Internet to gain access to operating systems, disrupting network services, or controlling network paths [2, 3]. Therefore, safeguarding information and ensuring privacy necessitates vigilant monitoring through defensive measures to detect and prevent unauthorised access to sensitive files. However, many challenges arise in designing and developing adaptable and effective methods for vulnerability detection, such as a great interest in using deep learning algorithms to improve cybersecurity, due to the use of the above-mentioned machine learning algorithms to address and detect security issues. Software vulnerabilities. Discovering software vulnerabilities is of paramount importance in today's world, as threats are becoming increasingly common. With new threats emerging all the time, the field of software vulnerability detection has become more challenging [4, 5]. Intrusion detection methods are classified into three main categories: signature-based, anomaly-based, and hybrid methods. Anomaly-based intrusion detection employs several techniques, including knowledge-based detection, statistical methods, and machine learning algorithms [6]. Machine learning algorithms encompass various types, such as supervised, unsupervised, and semi-supervised learning. Despite their differences, these techniques share a common principle: they rely on intrusion data, process it, classify it, and detect intrusions [7, 8]. Machine learning algorithms have been applied in various fields, including intrusion detection, where models are created and linked to algorithms to enhance dataset features. The primary goal of using machine

\*Corresponding author. Email: [ss.aa.cs@uomustansiriya.edu.iq](mailto:ss.aa.cs@uomustansiriya.edu.iq)

learning techniques is to develop more accurate identifiers and features to detect intrusions in frequently attacked network datasets [9, 10].

Regarding current intrusion detection systems, there is a research gap regarding the effectiveness and flexibility of the type of attack classification, whether binary or multi-class. On the other hand, intrusion detection systems work on large data sets that require pre-processing and at the same time selecting improved features to facilitate the detection process [11]. To be able to analyse the network intrusion in terms of whether it is a malicious or not. Therefore, in terms of the type of attack related to its source in the network traffic, we proposed a system that detects network intrusion through classification tools for the data set specific to this type of attack. The Corona Virus optimization Algorithm was adopted as one of the methods that have not been addressed in previous works specialized in this field in terms of improving the features of data sets specific to detecting network intrusion. This research can contribute to enhancing cybersecurity measures, protecting sensitive information, and ensuring the stability of network systems by using machine and deep learning models in binary and multi-classification.

The proposed approach provides better security and privacy for network systems by detecting the intrusion as malicious or secure and then indicating the exact type of intrusion if malicious. The features of the NSL-KDD dataset are improved by using the new Corona Virus algorithm in such a field.

This paper is structured as follows: section 2 presents the network intrusion detection methods, section 3 analyses a different of previous works related to the topic of the manuscript, section 4 presents the most common data optimization methods, section 5 is divided into subsections for the proposed method, while section 6 presents the results, compares the results with previous works, and discusses the results, finally, the conclusions are presented in section 7 of the paper.

## 2. NETWORK INTRUSION DETECTION METHODS

Network intrusion detection systems comprise a structure with several subcomponents, each performing integrated tasks related to intrusion detection [12]. Figure 1 illustrates the intrusion detection methods.

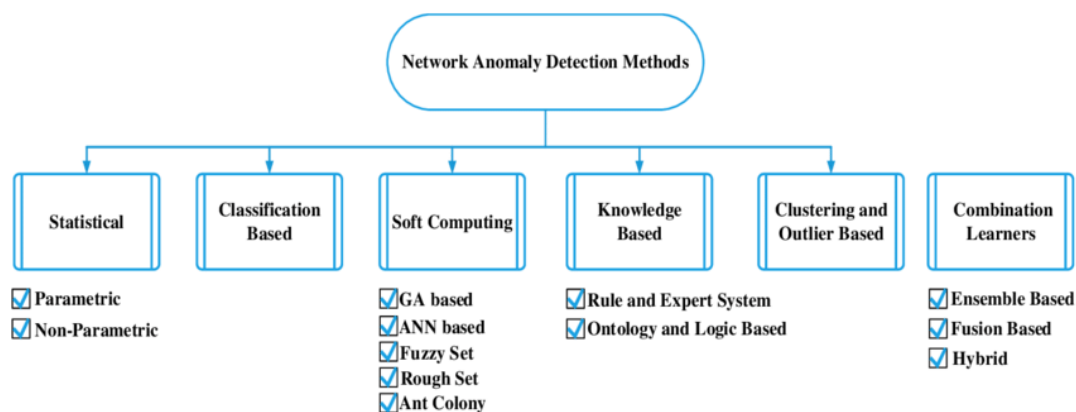


Fig. 1. Classification of intrusion detection methods [11].

Classification-based methods are crucial for detecting intrusions, especially when distinguishing between normal and attack cases, regardless of attack type. However, multi-class intrusion detection methods gain greater importance when balancing normal and aggressive attack classes [13].

## 3. LITERATURE REVIEW

Previous studies on intrusion detection have focused on machine learning algorithms. Table I summarises related research. The following is a review of literary works closely related to the research topic, particularly the use of optimisation methods with machine learning algorithms in intrusion detection.

Jothi and Pushpalatha (2023) [14] proposed a deep learning algorithm to design a system for detecting intrusion attempts in Internet of Things (IoT) networks. They analysed various features of IoT network nodes and used the CIDDS-001 and UNSWNB15 datasets to evaluate the proposed system's accuracy in detecting intrusions. The system achieved 95.20% accuracy with the CIDDS-001 dataset.

Sivamohan and Sridhar (2023) [15] designed an intrusion detection system using machine learning algorithms for classification. They employed the flock optimisation algorithm for feature optimisation to achieve the best accuracy. The

NSL-KDD dataset underwent pre-processing and was then analysed using the Interpretable Model-Agnostic Explanations LIME algorithm. The proposed method achieved 98.2% accuracy with the optimised features.

Alhayali et al. (2021) [16] proposed an intrusion detection method using the Rao optimisation tool to enhance features for machine learning algorithms, such as maximum learning machines and support vector machines. They used the KDDCup 99 and CICIDS2017 datasets to evaluate the method, achieving 100% and 97% accuracy, respectively, with the SVM algorithm and Rao optimisation tool.

Stiawan et al. (2020) [17] utilised intrusion detection system IDS generation methods to improve features from the ITD-UTM dataset for attack and intrusion detection, training classifiers with these improved features. They employed supervised methods in artificial intelligence, such as decision trees (J48) and the default network. The proposed system's virtual network achieved up to 85.3% accuracy in intrusion detection.

Jiang et al. (2020) [18] proposed a network intrusion detection system based on a convolutional neural network (CNN) model with a bi-directional long-term memory (BiLSTM) optimisation method. They analysed the NSL-KDD dataset using BiLSTM combined with CNN, achieving 83.58% classification accuracy for multi-class detection.

Wei et al. (2019) [19] designed a particle swarm optimisation model for initial data optimisation, followed by search probability to solve the initial optimisation based on gene intersection factors. The main intrusion detection system used a deep belief network (DBN) based on the initial optimisation results. The proposed method proved effective, reducing the average time for intrusion detection to 0.24 and achieving 82.36% accuracy with the DBN method.

Al Tobi and Duncan (2019) [20] studied how adapting the discrimination threshold to model predictions, particularly for estimated traffic, improves intrusion detection performance. They used machine learning algorithms such as C5.0, Random Forest, and Support Vector Machine, employing the STA2018 dataset. The results highlighted the importance of adapting thresholds to enhance intrusion detection accuracy, with Random Forest achieving 94.26% accuracy.

Alhayali et al. (2021) [16] proposed a method to improve intrusion detection by combining the Rao optimisation tool with various machine learning algorithms, such as maximum learning machines and support vector machines. They analysed and evaluated the proposed method using the KDDCup 99 dataset and CICIDS 2017 datasets, achieving 100% and 97% accuracy, respectively, with the SVM method and Rao optimisation.

TABLE I. SUMMARY OF RELATED RESEARCH

Author	Technique	Dataset	Optimisation method	Classification type	Limitation
Jothi and Pushpalatha 2023 [14]	LSTM	CIDDS-001 and UNSWNB15	Without optimization	Multi-class	It uses an LSTM network which generates overfitting and misalignment with long data.
Sivamohan and Sridhar 2023 [15]	LIME	NSL-KDD	flock optimisation algorithm	Multi-class	Flock optimisation algorithm needs to adapt effectively to different constraints. This is not suitable for the type of data used.
Alhayali et al. 2021 [16]	support vector machine	KDDCup 99 and CICIDS 2017	Rao Optimisation Tool	Multi-class	Using the Rao Optimisation Tool requires data processing first, which is not supported by the author.
Stiawan, Deris et al. 2020 [17]	DT, J48	ITD-UTM	Optimal IDS generation	Binary class	It was able to detect malicious and harmless activities without revealing the type of intrusion.
Jiang, Kaiyuan, et al. 2020 [18]	CNN, BiLSTM	NSL-KDD	Without optimization	Multi-class	The author's proposal is to use the same dataset as in this manuscript without improving its features.
Wei et al. 2019 [19]	Deep Belief Network	NSL-KDD	Particle swarm optimisation	Multi-class	Use of Particle swarm optimisation algorithm which requires real-time data even though the dataset is pre-collected
Al Tobi and Duncan 2019 [20]	C5.0, RF, and SVM	STA2018	adapting threshold	Binary class	It was able to detect malicious and harmless activities without revealing the type of intrusion.

#### 4. OPTIMISATION METHODS

Optimisation techniques and methods are considered influential factors in improving the accuracy of classification algorithms and reducing the complexity of detection and classification problems in record time. Effective data extraction and optimal selection enable better data preparation when designing and creating classification models, especially for data containing errors or unnecessary spaces [21, 22]. The role of the optimiser is to make adjustments during the search and update process until the required values, deemed optimal, are achieved. The optimal solution involves finding the minimum and maximum solutions and employing them to reach the goal [21, 23]. In deep learning, optimisation is crucial for obtaining optimal solutions and features in a neural network [24, 25]. Most machine learning problems can be solved by rearranging or updating the data to suit the nature of the problem and the algorithm's mechanism. Optimisation methods vary according to machine learning techniques, necessitating the continuous search for the best improvement methods to obtain optimal solutions [26].

Given the multitude of optimisation algorithms, we will discuss a subset that relies on population-based principles, focusing on methods chosen to enhance search techniques. These include genetic algorithms, swarm optimisation algorithms, and the modern coronavirus algorithm. Genetic algorithms, a subset of evolutionary algorithms, are primarily used to find solutions to optimisation problems. They simulate the process of natural selection [27]. Particle swarm optimisation is a random algorithm that relies on the intelligent behaviour of individuals and groups of animals, such as flocks of birds or schools of fish, to find optimal solutions [28, 29]. The coronavirus algorithm, inspired by the spread of the virus and the probability of infection, uses terms like infected person, healthy person, distancing, death, and complete recovery [23, 30].

The coronavirus algorithm was chosen for its ability to determine input parameters based on disease statistics and updated infection information, avoiding the use of random values that may cause the algorithm to fail to reach an optimal solution [31]. The steps for the coronavirus optimisation algorithm are summarised as follows:

- Definition of the population set, including all vectors in the dataset and their features.
- Defining the first individual in the population as the initial infection, assigning it a value of 0, which can be chosen randomly.
- The spread of the disease occurs through the first infected person, with the following possible outcomes:
  - 1- The infected person dies, preventing further transmission.
  - 2- Transmission of the disease to new individuals, with a 10 percent probability of spreading.
  - 3- Isolation of the infected person to prevent further spread, with a 50 percent probability of isolation.
- Updating the data by categorising the group into:
  - 1- Deaths, with a probability of 0.05.
  - 2- Complete recovery of the infected person.
  - 3- Infection of new individuals, with infection rates between 0 and 5 if the disease spreads naturally.
- The algorithm's stopping criteria depend on a complete recovery or death. The current population number must be less than the original population, with a specified number of repetitions or a time period for stopping.

Using the coronavirus optimisation technique will yield satisfactory results in selecting the most important vectors before the profit-making stage in a specialised dataset by network layers and detecting portals.

#### 5. PROPOSED METHODOLOGY

The proposed system detects network intrusions based on the optimal solution, updating the dataset according to the previously mentioned coronavirus working principle. As shown in Figure 2, the system includes three main phases: pre-processing, which involves data normalisation, and searching for optimal solutions among the dataset features, modified after each iteration in the training stage of the proposed models. The training phase includes two models: the first uses CNN to classify the attack type, resulting in multiple types such as 'Denial of Service', 'Probe', 'User to Root', and 'Remote to Local'. The second model uses a decision tree to determine whether the attack classified by CNN is a normal or abnormal intrusion. Finally, the system's performance is evaluated and tested to determine its accuracy in detecting the type and predicting the nature of the intrusion.

## 5.1 Dataset Collection

The dataset used to evaluate the classification models is the state-of-the-art NSL-KDD, developed from the KDD'99 dataset by the University of New Brunswick, NSL-KDD and CICIDS2017 datasets for intrusion detection and anomaly detection in IoT networks [11]. The NSL-KDD dataset includes four main classes, each potentially leading to an effective and malicious attack or being normal. The attack types are denial of service, probe, user to root, and remote to local. Table II shows the dataset, including the numbers of each attack type and its classes, normal or abnormal.

TABLE II. NSL-KDD DATASET DETAILS

Attack type	Denial of Service	Probe	User to Root	Remote to Local
Normal	15958	22903	25001	24983
Abnormal	9234	2289	180	209

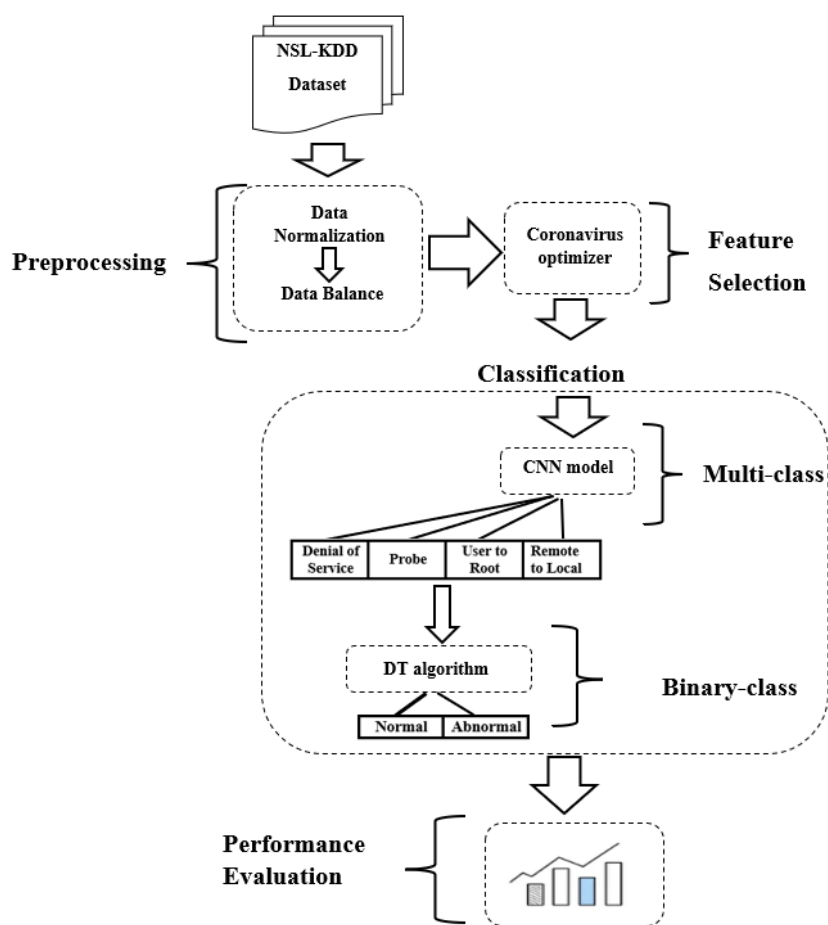


Fig. 2. An architecture of the proposed model.

## 5.2 Preprocessing Operations

### 5.2.1 Data normalisation

The most important preprocessing step is to normalise the features in the dataset for different types of network attacks. This step involves converting the dataset content, represented by vectors, into categories with numerical values, making it easier to train the classification model more efficiently.

In this paper, the Standard-Scaler method was used to transform the dataset into standard normal distributions.

The main task of Standard-Scaler is to normalise features to have a mean of 0 and a standard deviation of 1, resulting in a standard normal distribution. The following equation shows normalisation using Standard-Scaler:

$$X_{sd} = \frac{X - \mu}{\sigma} \quad (1)$$

Here  $X$  refers to the original feature, the average feature is indicated by  $\mu$ , and the deviation is represented by  $\sigma$ .

### 5.2.2 Data balance

Datasets like the one used in this work often contain class imbalance issues that occur during data creation and compilation. It is important to address this problem using rebalancing methods such as the SMOTE technique.

This technique involves identifying minority classes relative to the rest of the dataset and creating new instances for these minority classes.

For every  $X_i$  belonging to the minority class set:

Determine the nearest neighbour of  $X_i$  by calculating the distance between  $X_i$  and all members of the minority class set using Euclidean distance.

$$D(X_i, X_j) = \sqrt{\sum_{l=1}^n (X_{il} - X_{jl})^2} \quad (2)$$

Let  $X_j$  be a neighbor of  $X_i$ . Artificial instances are created for each neighbour called  $X_n$  based on the following equation:

$$X_n = X_i + \lambda \times (\text{random.neighbor} - X_i) \quad (3)$$

$\lambda$  is considered a random value greater than 0 and less than 1, which is the nearest neighbour.

## 5.3 Classification Models

Initially, the optimal feature selection is based on the coronavirus algorithm. Since feature improvement methods are crucial for obtaining data with enhanced features, the coronavirus algorithm was chosen to improve the features and use them as input for the classification stage and detecting the type of intrusion attack.

### 5.3.1 First model-multi-class

The CNN model is used to classify the dataset into four classes: Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L), based on the coronavirus algorithm for optimal feature selection. The decision tree algorithm is then used to determine whether the attack classified by CNN is a normal or abnormal class according to the type of attack in the four outputs.

The structure of the proposed CNN consists of several layers, starting with the input layer, followed by hidden layers, and finally the output layer for the four types of attacks.

Hidden layers consist of a number of filters ranging from 32 to 256, with the number of filters varying depending on the features and their complexity in each layer. The max pooling layers use ReLU as the activation function. To ensure that all grid cells are not involved simultaneously, the ReLU function in (4) is ideal, as it is non-linear and does not stop working until the transformation is linear and equal to zero [32][34].

$$f(x) = \max(0, x) = \begin{cases} x_i, & \text{if } x_i \geq 0 \\ 0, & \text{if } x_i < 0 \end{cases} \quad (4)$$

Since classification involves more than two classes, we usually need an active multi-class function in the output layer, such as Softmax, as shown in (5) [33][35].

$$f(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \quad (5)$$

### 5.3.2 Second model - binary class

The DT algorithm will be used to determine the type of attack. The inputs to these algorithms are the output of the CNN model, which identifies the type of attack, along with the features for each type in the dataset used.

The non-linear sigmoid activation function, represented by (6), is typically used for binary classification types [33].

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (6)$$

## 6. EXPERIMENTAL RESULTS

After conducting numerous attempts to find the optimal values for the parameters within the CNN classification model, Table III and Figure 2 show the change in parameter values and their effect on the model's accuracy, relying on the learning rate, number of epochs, and kernel size in the layer.

TABLE III. RESULTS WITH DIFFERENT PARAMETERS

No.	Epochs	Kernel size	Learning rate	Accuracy
1	10	5	0.001	99.22%
2	22	3	0.001	99.15%
3	15	3	0.001	99.28%
4	20	5	0.0001	99.21%
5	13	3	0.001	99.32%

The highest accuracy achieved by the proposed model was 99.32% at a learning rate of 0.001, with 13 epochs and a kernel size of 3, as shown in Figure 3.

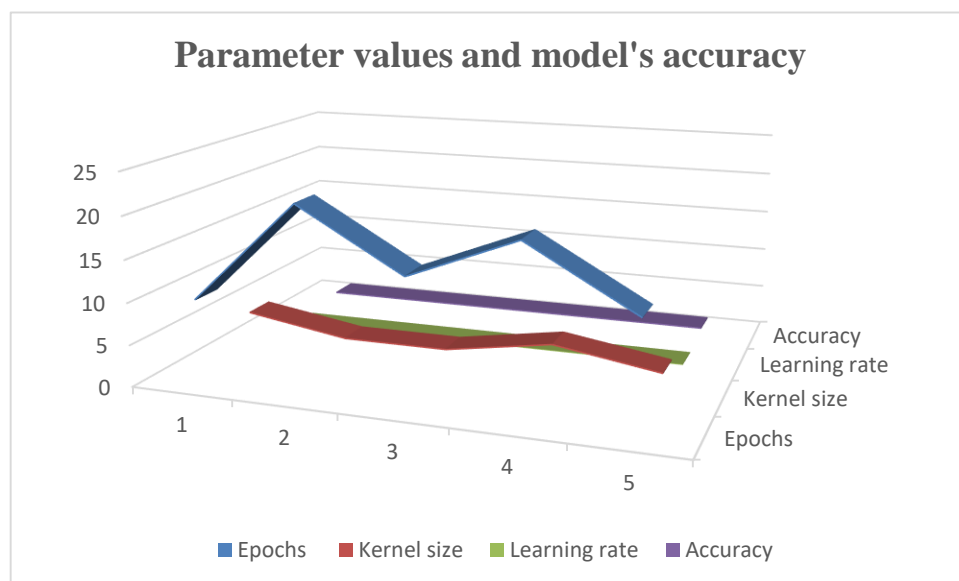


Fig. 3. The effect of parameter values on the accuracy curve.

The performance metrics for applying the CNN classifier to the dataset are shown in Table IV and Figure 4. The highest values for all performance metrics for the CNN classifier reach 100 percent for the Remote to Local attack type.



TABLE IV. PERFORMANCE MEASURES OF CNN IN MULTI-CLASS

Performance Measures	Classes			
	<i>Denial of Service</i>	<i>Probe</i>	<i>User to Root</i>	<i>Remote to Local</i>
Precision	0.887	0.983	1	1
Recall	0.995	0.755	0.988	1
F-measure	0.962	0.926	0.986	1

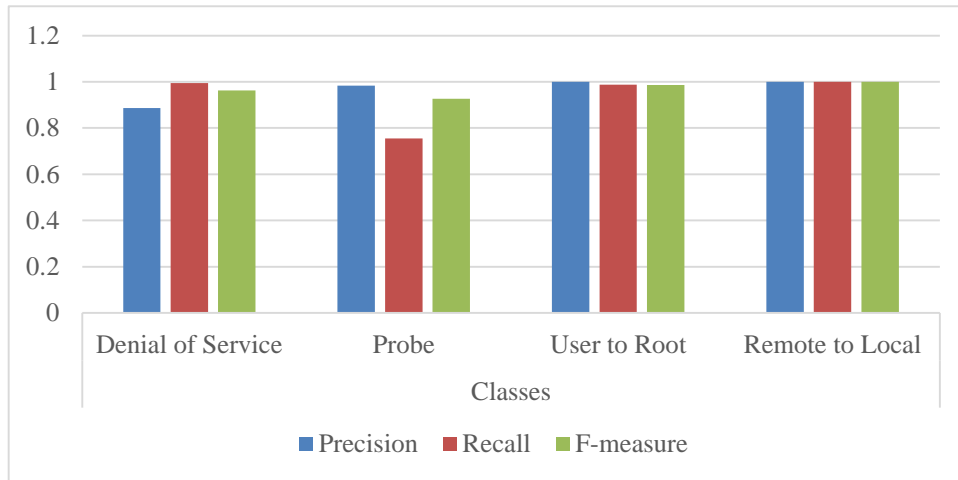


Fig. 4. Performance measures of CNN in multi-class.

In the second stage of intrusion detection, the various types of data are divided into normal or abnormal attacks, which in any case come from one of the previous types as primary attacks. To determine which type has an abnormal attack pattern or a normal pattern, the decision tree classifier was trained using the classified outputs from the CNN model.

The performance of the decision tree upon testing appeared as shown in Table V and Figure 5.

TABLE V. PERFORMANCE MEASURES OF DT IN BINARY CLASS

Class	Precision	Recall	F-measure
Normal	0.8490	0.8767	0.8538
Abnormal	0.8864	0.8678	0.8864

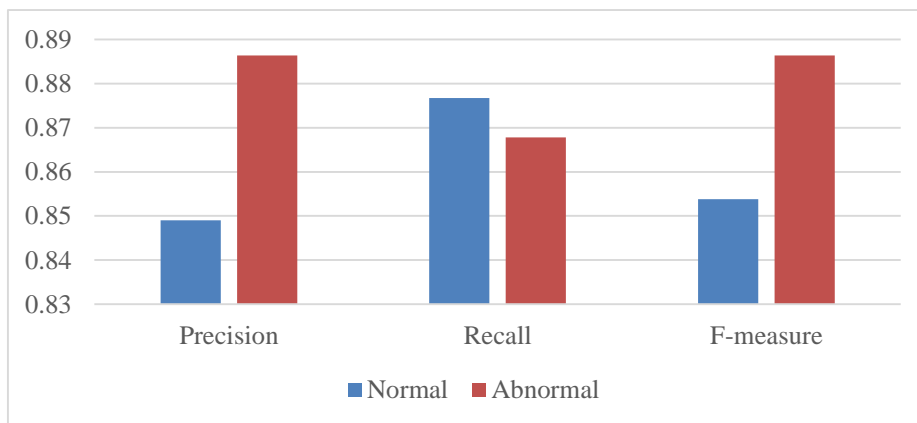


Fig. 5. Performance measures of DT in binary class.



## 6.1 Comparison with Previous Works

When the proposed system is compared with previous related works in terms of the NSL-KDD dataset type, optimisation methods, and classification type, as shown in Table VI. It was found that the current system excelled in several aspects. The most important of these are classification accuracy, the use of both multi-class and binary classification, and the use of an optimisation algorithm that had not been previously employed in the field of intrusion detection. Previous methods often focused solely on determining the type of attack, regardless of whether it was effective or normal.

TABLE VI. COMPARISON WITH RELATED WORKS

Author	Technique	Classification type	Optimisation method	Accuracy
Sivamohan and Sridhar 2023 [15]	LIME	Multi-class	Flock optimisation algorithm	98.2%
Jiang, Kaiyuan, et al. 2020 [18]	CNN	Multi-class	BiLSTM	83.58%
Wei et al. 2019 [19]	Deep belief network	Multi-class	particle swarm optimisation	82.36%
Proposed model	CNN DT	Multi-class Binary class	Coronavirus	99.3% 0.8864

## 6.2 The Discussion

The problem of network intrusion detection is still present due to the growing of data transmission technologies over the network - therefore, we trained classification models on the NSL-KDD dataset. Most of the authors of previous works relied on one of the types of classification, either binary or multiclass classification.

The manuscript proposal does not only depend on the combination of both binary and multiclass classification types, but it mainly depends on optimizing the features of the NSL-KDD dataset using the Corona Virus optimization algorithm instead of using previously known optimization algorithms such as PSO, WSA, TBO, and DBO. The resulting optimization solution works as a network structure for the intrusion classification model. The classification also combines binary and multivariate in a different way from previous work. Although the proposed system achieves an accuracy of up to 99.3% for the CNN model in multi-classification and 88.64% for binary classification, the work is limited to the NSL-KDD dataset and needs to be adapted for real-time intrusion detection of illegal network intrusion.

## 7. CONCLUSION

In this paper, a new approach was proposed to enhance the features of the dataset for intrusion detection systems by adopting the coronavirus optimisation algorithm. The optimisation algorithm was based on the principle of the spread of the coronavirus disease 19 and how to obtain the best data for its ability to persist because it carries the best features. Most previous studies have adopted either binary or multi-class classification, in this manuscript we have adopted both types of binary and multi-class classification instead of using the Corona Virus Optimization Algorithm which has not been used in any previous work in the field of study. The proposed models were trained on NSL-KDD datasets, with CNN as a multi-class classifier and DT as a binary classifier. The system architecture is designed to automatically learn the features of the four attack types, which are further divided into two categories: natural and unnatural. According to system performance measures and testing, it proved superior in detecting the attack and its type compared to previous works that used different improvement methods. For better results, it is crucial to focus on developing intrusion detection systems that can respond to threats as they occur. Future research could focus on optimisation techniques to reduce latency in intrusion detection systems and use training stall metrics within the neural network. It is important to note that the work was not based on real-time intrusion detection, but it may be used in the future for real-time data on the network after likened it for analysing the attack, then adding it to a new dataset.

## Conflicts Of Interest

The authors declare no conflicts of interest.

## Funding

No funding.

## Acknowledgment

The authors would like to express their thanks to the Department of Computer Science, College of Science, Mustansiriyah University. (<https://uomustansiriyah.edu.iq/e-newsite.php>), for supporting this work.

## References

- [1] A. A. Abdulhameed, R. J. Al-Azawi, and B. M. Al-Mahdawi, "Modeling web security analysis attacks with cysemol tool," *Al-Mustansiriyah Journal of Science*, vol. 31, no. 3, pp. 101-109, 2020.
- [2] S. Mishra, R. Sagban, A. Yakooob, and N. Gandhi, "Swarm intelligence in anomaly detection systems: an overview," *International Journal of Computers and Applications*, vol. 43, no. 2, pp. 109-118, 2021.
- [3] H. Debar, "An introduction to intrusion-detection systems," *Proceedings of Connect*, vol. 2000, 2000.
- [4] R. Tahri, Y. Balouki, A. Jarrar, and A. Lasbahani, "Intrusion detection system using machine learning algorithms." p. 02003.
- [5] J. Du, K. Yang, Y. Hu, and L. Jiang, "NIDS-CNNLSTM: Network intrusion detection classification model based on deep learning," *IEEE Access*, vol. 11, pp. 24808-24821, 2023.
- [6] O. Elnakib, E. Shaaban, M. Mahmoud, and K. Emara, "EIDM: Deep learning model for IoT intrusion detection systems," *The Journal of Supercomputing*, vol. 79, no. 12, pp. 13241-13261, 2023.
- [7] G. Abdelmoumin, D. B. Rawat, and A. Rahman, "On the performance of machine learning models for anomaly-based intelligent intrusion detection systems for the internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4280-4290, 2021.
- [8] H. A. Mahmood, and S. H. Hashem, "Network intrusion detection system (NIDS) in cloud environment based on hidden Naïve Bayes multiclass classifier," *Al-Mustansiriyah Journal of Science*, vol. 28, no. 2, pp. 134-142, 2018.
- [9] B. Kaushik, R. Sharma, K. Dhama, A. Chadha, and S. Sharma, "Performance evaluation of learning models for intrusion detection system using feature selection," *Journal of Computer Virology and Hacking Techniques*, vol. 19, no. 4, pp. 529-548, 2023.
- [10] M. H. Al-Tai, B. M. Nema, and A. Al-Sherbaz, "Deep learning for fake news detection: Literature review," *Al-Mustansiriyah Journal of Science*, vol. 34, no. 2, pp. 70-81, 2023.
- [11] Y. L. Khaleel, M. A. Habeeb, A. Albahri, T. Al-Quraishi, O. Albahri, and A. Alamoodi, "Network and cybersecurity applications of defense in adversarial attacks: A state-of-the-art using machine learning and deep learning methods," *Journal of Intelligent Systems*, vol. 33, no. 1, pp. 20240153, 2024.
- [12] E. İ. Uysal, G. Demircioğlu, G. Kale, E. Bostanci, M. S. Güzel, and S. N. Mohammed, "Network Anomaly Detection System using Genetic Algorithm, Feature Selection and Classification." pp. 1-5.
- [13] N. Moustafa, J. Hu, and J. Slay, "A holistic review of network anomaly detection systems: A comprehensive survey," *Journal of Network and Computer Applications*, vol. 128, pp. 33-55, 2019.
- [14] B. Jothi, and M. Pushpalatha, "WILS-TRS—A novel optimized deep learning based intrusion detection framework for IoT networks," *Personal and Ubiquitous Computing*, vol. 27, no. 3, pp. 1285-1301, 2023.
- [15] S. Sivamohan, and S. Sridhar, "An optimized model for network intrusion detection systems in industry 4.0 using XAI based Bi-LSTM framework," *Neural Computing and Applications*, vol. 35, no. 15, pp. 11459-11475, 2023.
- [16] R. A. I. Alhayali, M. Aljanabi, A. H. Ali, M. A. Mohammed, and T. Sutikno, "Optimized machine learning algorithm for intrusion detection," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 1, pp. 590-599, 2021.
- [17] D. Stiawan, A. Heryanto, A. Bardadi, D. P. Rini, I. M. I. Subroto, M. Y. B. Idris, A. H. Abdullah, B. Kerim, and R. Budiarto, "An approach for optimizing ensemble intrusion detection systems," *Ieee Access*, vol. 9, pp. 6930-6947, 2020.
- [18] K. Jiang, W. Wang, A. Wang, and H. Wu, "Network intrusion detection combined hybrid sampling with deep hierarchical network," *IEEE access*, vol. 8, pp. 32464-32476, 2020.
- [19] P. Wei, Y. Li, Z. Zhang, T. Hu, Z. Li, and D. Liu, "An optimization method for intrusion detection classification model based on deep belief network," *Ieee Access*, vol. 7, pp. 87593-87605, 2019.
- [20] A. M. Al Tobi, and I. Duncan, "Improving intrusion detection model prediction by threshold adaptation," *Information*, vol. 10, no. 5, pp. 159, 2019.
- [21] E. O. Abiodun, A. Alabdulatif, O. I. Abiodun, M. Alawida, A. Alabdulatif, and R. S. Alkhalwaldeh, "A systematic review of emerging feature selection optimization methods for optimal text classification: the present state and prospective opportunities," *Neural Computing and Applications*, vol. 33, no. 22, pp. 15091-15118, 2021.

- [22] D. Yi, J. Ahn, and S. Ji, "An effective optimization method for machine learning based on ADAM," *Applied Sciences*, vol. 10, no. 3, pp. 1073, 2020.
- [23] H. Emami, "Anti-coronavirus optimization algorithm," *Soft Computing*, vol. 26, no. 11, pp. 4991-5023, 2022.
- [24] M. A. Al-Betar, Z. A. A. Alyasseri, M. A. Awadallah, and I. Abu Doush, "Coronavirus herd immunity optimizer (CHIO)," *Neural Computing and Applications*, vol. 33, no. 10, pp. 5011-5042, 2021.
- [25] R. Sun, "Optimization for deep learning: theory and algorithms," *arXiv preprint arXiv:1912.08957*, 2019.
- [26] S. Sun, Z. Cao, H. Zhu, and J. Zhao, "A survey of optimization methods from a machine learning perspective," *IEEE transactions on cybernetics*, vol. 50, no. 8, pp. 3668-3681, 2019.
- [27] P. Misra, and A. S. Yadav, "Impact of preprocessing methods on healthcare predictions."
- [28] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," *Soft computing*, vol. 22, pp. 387-408, 2018.
- [29] F. Pourpanah, R. Wang, C. P. Lim, X.-Z. Wang, and D. Yazdani, "A review of artificial fish swarm algorithms: Recent advances and applications," *Artificial Intelligence Review*, vol. 56, no. 3, pp. 1867-1903, 2023.
- [30] F. Martínez-Álvarez, G. Asencio-Cortés, J. F. Torres, D. Gutiérrez-Avilés, L. Melgar-García, R. Pérez-Chacón, C. Rubio-Escudero, J. C. Riquelme, and A. Troncoso, "Coronavirus optimization algorithm: a bioinspired metaheuristic based on the COVID-19 propagation model," *Big data*, vol. 8, no. 4, pp. 308-322, 2020.
- [31] Y. Yuan, Q. Shen, S. Wang, J. Ren, D. Yang, Q. Yang, J. Fan, and X. Mu, "Coronavirus mask protection algorithm: A new bio-inspired optimization algorithm and its applications," *Journal of Bionic Engineering*, vol. 20, no. 4, pp. 1747-1765, 2023.
- [32] N. Gupta, P. Bedi, and V. Jindal, "Effect of activation functions on the performance of deep learning algorithms for network intrusion detection systems." pp. 949-960.
- [33] A. D. Rasamoelina, F. Adjailia, and P. Sinčák, "A review of activation function for artificial neural network." pp. 281-286.
- [34] F. K. H. Mihna, M. A. Habeeb, Y. L. Khaleel, Y. H. Ali, and L. A. E. Al-saeedi, "Using Information Technology for Comprehensive Analysis and Prediction in Forensic Evidence," *Mesopotamian J. CyberSecurity*, vol. 2024, pp. 4–16, 2024, doi: 10.58496/mjcs/2024/002.
- [35] L. A. E. Al-saeedi, F. J. Shakir, F. K. Hasan, G. G. Shayea, Y. L. Khaleel, and M. A. Habeeb, "Artificial Intelligence and Cybersecurity in Face Sale Contracts: Legal Issues and Frameworks," *Mesopotamian J. CyberSecurity*, vol. 4, no. 2, pp. 129–142, 2024.