Research Article

Waad Falah Kamil* and Imad Jasim Mohammed

Deep learning model for intrusion detection system utilizing convolution neural network

https://doi.org/10.1515/eng-2022-0403
received October 26, 2022; accepted January 10, 2023

Abstract: An integral part of any reliable network security infrastructure is the intrusion detection system (IDS). Early attack detection can stop adversaries from further intruding on a network. Machine learning (ML) and deep learning (DL) techniques to automate intrusion threat detection at a scale never previously envisioned have snowballed during the past 10 years. Researchers, software engineers, and network professionals have been encouraged to reconsider the use of ML techniques, notably in cybersecurity. This article proposes a system for detecting intrusion with two approaches, the first utilizing a proposed hybrid convolutional neural network (CNN) and Dense layers. The second utilizes naïve Bayes (NB) ML techniques and compares the two approaches to determine the best detection accuracy. The preprocessing of network data is necessary. The suggested technique is evaluated using the UNSW-NB15 Dataset to create a reliable classifier and an effective IDS. The experimental results for the proposed CNN-dense classifier outperformed the ML and DL models. CNN has a 99.8% accuracy rate compared to previous studies. At the same time, the Gaussian naïve Bayes, which is considered the best among the ML-utilized classifiers, yielded an 83% accuracy rate.

Keywords: intrusion detection system, machine learning, deep learning, convolutional neural network, naïve Bayes, UNSW-NB15

1 Introduction

Since the Internet's establishment, information systems that utilize or are based on it have progressed dramatically, like the World Wide Web. Most cyberattacks are launched via the notoriously insecure Internet [1]. One of the most significant issues confronting security management system developers is ensuring the protection and privacy of big data, particularly in light of the widespread utilization of Internet networks and the explosive development in the amount of data created from various sources [2]. Attempts to breach or circumvent the confidentiality, integrity, and availability of security processes that protect networks and computer resources are called intrusions [3]. As an active security mechanism, intrusion detection systems (IDSs) are a potent tool and a crucial part of the infrastructure that ensures the safety of the networks we rely on daily, which can be hardware or software. IDS monitors and analyzes data as it travels through computers and networks to detect security problems [4]. Misuse and anomaly detection are the two fundamental methodologies utilized by IDSs to examine events and identify attacks [5].

The challenge is to create methods for identifying threats through deep learning (DL) approaches to improve the system's efficacy and accuracy while decreasing the number of false alarms with little computing effort [6]. Big data presents a significant challenge to IDSs due to its volume, diversity, and velocity. "Big data" refers to information that is difficult to handle, store, or manipulate utilizing typical methods [7]. The term "big" refers to the amount of data acquired from various sources, which has grown significantly in recent years [8]. Machine learning (ML) and DL methods may be divided into four main groups, namely supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning, as illustrated in Figure 1. The first two categories obtain the majority of the intrusion detection research that has been published in the literature [9].

This work proposes a system to detect network intrusion based on DL and ML techniques. A convolutional neural network (CNN) and naïve Bayes (NB) were utilized for classification. These suggested approaches are applied to the UNSW-NB15 Dataset, which contains a group of common and updated attacks.

This article is organized as follows: Section 2 discusses the related works. Section 3 briefly describes the dataset,

*** Corresponding author: Waad Falah Kamil,** Department of Computer Science, University of Baghdad, Baghdad, Iraq,
e-mail: pcwaad@gmail.com
Imad Jasim Mohammed: Department of Computer Science, University of Baghdad, Baghdad, Iraq

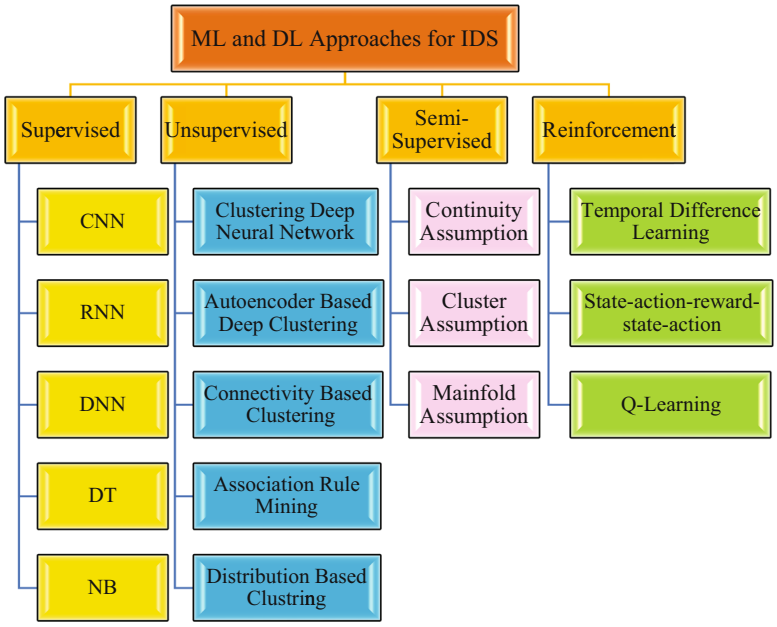

Figure 1: Classification of ML and DL algorithms for detecting network intrusions [9].

preprocessing, and classification techniques used. Section 4 gives details of the evaluation metrics. Section 5 contains the findings and discussion. Section 6 ends with the conclusion and future work.

2 Related works

Computer networks are implementing the most recent technologies as technology and current approaches improve, significantly impacting the intensity of attacks. As a result, the UNSW-NB15 Dataset was utilized with specific attention paid to the modern forms of attack. Research utilizing the UNSW-NB15 Dataset has not yet reached its full potential. However, certain studies that made use of datasets are summarized here. The relevant research is compared and summarized in Table 1.

For studies based on ML techniques, Kumar et al. [10] developed a calcification-based integrated network intrusion detection system (NIDS) that utilized clusters generated by IG's feature selection approach and the *k*-means algorithm in conjunction with decision tree (DT) algorithm. The RTNITP18 Dataset, with 22 features and four types of network assaults from the UNSW-NB15 Dataset, was used as a test dataset to assess the efficacy of the proposed model. Compared to the DT C5 model, whose accuracy was 90.74%, the proposed model was only 84.83%.

Kasongo and Sun [11] presented the NIDS technology, which combines the feature selection method of the XGBoost algorithm with five classification techniques: logistic regression (LR), k-nearest neighbors (KNN), artificial neural network (ANN), DT, and support vector machine (SVM). The UNSW-NB15 Dataset was classified using binary and multiclass techniques. The maximum accuracy of multiclass classification was 82.66%, whereas the KNN classifier accuracy was 96.7%.

Kumar et al. [12] recommended the unified intrusion detection system (UIDS) using the UNSW-NB15 Dataset. The ruleset (R) utilized to create the proposed UIDS model was taken from various DT models, including *k*-means clustering and IG's feature selection method. The model was also trained with various methods, including C5, neural networks, and support vector machines. For this reason, the suggested model achieved a higher accuracy (88.92%) than any competing methods. Other algorithms have higher accuracy, such as C5, neural network, and SVM (89.76, 86.7, and 78.77%, respectively).

Shushlevska et al. [13] implemented IDS depending on the UNSW-NB15 Dataset. The dataset has been tested and trained for nine distinct class assaults. Utilizing four ML methods, the UNSW-NB15 Dataset was efficiently split into network traffic of ordinary records and attack logs. The classification utilizing Random Forest (RF) is more efficient than with NB, LR, and DT, according to the study of the ML models for each of the approaches. The RF classifier offers an accuracy value of 95.9% in accordance with the results attained.

For studies relying on DL approaches, P. Wu and H. Guo [14] suggested using a LuNet model to discover

Table 1: Summary of existing studies on IDS based on ML and DL techniques

Ref.	Datasets	Algorithm	Best accuracy (%)	Limitations
ML techniques				
[10]	UNSW-NB15	DT models (CART, CHAID, QUEST, and C5)	90.74	This study predicted just four of the nine categories in the UNSW-NB15 Dataset. Furthermore, this study has not addressed the issue of class inequality.
[11]	UNSW-NB15	LR, *K*-nearest neighbors, DT, ANN, and SVM	96.76	This study did not address the issue of class inequalities. As a result, the model's accuracy is poor.
[12]	UNSW-NB15	Models of DT (C5, CHAID, CART, and QUEST)	88.92	The UNSW-NB15 Dataset comprises only four types of network attack categories predicted by research. Furthermore, this study has not addressed the issue of class inequality.
[13]	UNSW-NB15	RF, LR, NB, and DT	95.9	Feature selection methods must be applied to the UNSW-NB15 Dataset.
DL techniques				
[14]	UNSW-NB15	CNN and RNN	84.98	This research did not improve dealing with an RNN algorithm, although there are studies that worked on the same dataset and gave good results.
[15]	UNSW-NB15	CNN	93.5	The use of data preparation techniques in this study has not been done well.
[16]	UNSW-NB15	CNN	94.22	Feature, selection, or reduction methods do not apply to the UNSW-NB15 Dataset.
[17]	UNSW-NB15	CNN	92.10	This study must be expanded to detect multiclass classification intrusions.

intrusions on a large-scale network, which is a hierarchical recurrent neural networks (RNN) and CNN used on the NSL-KDD and UNSW-NB15 dataset. On the NSL-KDD dataset, the accuracy in binary classification obtained an average of 99.24%, and on the UNSW-NB15 dataset, it was 97.40%. On average, NSL-KDD performed with 99.05% accuracy while UNSW-NB15 performed with 84.98% accuracy in multiclass classification.

Mahalakshmi et al. [15] created the CNN DL technology to handle the challenge of detecting network infiltration. The CNN algorithm was trained to utilize the UNSW NB15 public dataset. In general, the dataset comprises binary types of "0" and "1" for normal and assault data. The experimental findings indicated that the suggested model has a maximum detection accuracy of 93.5%.

Singh et al. [16] suggested a brand-new wide deep transfer learning (TL) GRU model. A preprocessing procedure is created for multi-dimensional point data (multivariate time series) (UNSW-NB15). Wide deep is made up of a linear model, and the deep component is made up of Base-4GRU, TL-3GRU-1, or TL4GRU-2. According to the experimental findings, the suggested solution beats most of the current network intrusion detection strategies on ML, with an evaluation accuracy of 94.22% on the UNSW-NB15.

Almarshdi et al. [17] created an IDS architecture based on a CNN and LSTM model combination to identify security assaults in the IoT utilizing the UNSW-NB15 Dataset. The missing values in the dataset are eliminated based on the interpolation procedure. The suggested model was compared to the CNN model on a balanced and unbalanced dataset. Then, utilizing a balanced dataset, they compare the suggested model against DT and RF ML classifiers. On the balanced dataset, the suggested model outperformed CNN, DT, and RF with an accuracy of 92.10%.

3 Proposed methodology

The proposed system includes two methods for intrusion detection of networks, one utilizing DL and the other based on ML. In both cases, the data go through a preprocessing stage. Then the classification is done utilizing the two methods mentioned above, and the results of the two methods are compared. Figure 2 explains the overall diagram of the proposed system and the factors considered to build a hybrid model.

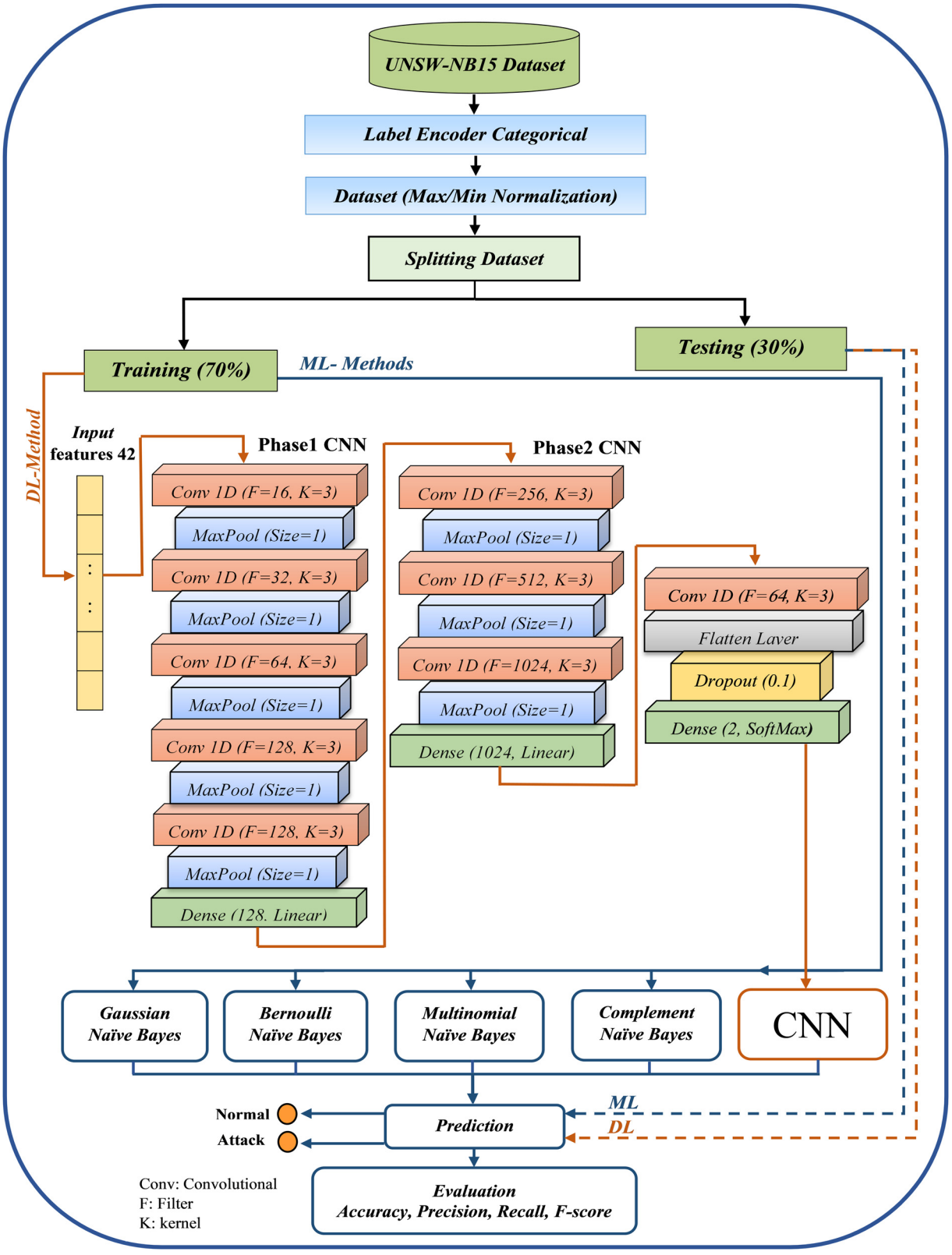

Figure 2: The proposed CNN-Dense classification framework vs ML.

3.1 A summary of the considerations for the proposed

- Early Stopping: Training should be halted when the validation error exceeds the minimum.
- Dropout: A regularization technique that is similar to training. Randomly ignoring some layer outputs forces nodes inside a layer to assume greater or lesser responsibility for the inputs.
- Adam's optimizer integrates the techniques for gradient descent, momentum, and Root Mean Squared Propagation. Every node in the network has its learning rate updated by Adam's optimizer, which lowers the overall error rate. It can process large datasets quickly and efficiently using less memory than other optimization methods and requiring less tuning.
- In the convolutional layers, we used a leaky rectified linear unit (ReLU) activation function. It has the same form as the ReLU, except that positive values close enough to zero will lead to zero. Avoid the dead node issue and do not have a vanishing gradient.
- In the medial dense layers, the linear activation function combines strong and close-to-strong features from the layers preceding it.
- In the last dense layer (classification), the SoftMax activation function is used for problems with more than one class. Methods take in a vector of the raw outputs of a neural network and give back a vector of probability scores.
- The last convolutional layer used a linear activation function with a small filter size to extract close-to-strong features from the layer preceding it.

3.2 Dataset description

The Cyber Range Lab at the Austrian Centre for Cyber Security uses raw network packets from the UNSW-NB15 Dataset to construct hybrid real-time normal operations and simulate current attack behavior using the IXIA PerfectStorm technology. Tcpdump stores 100 GB of raw traffic. Forty-nine characteristics, including the class label, are generated using the Argus and Bro-IDS tools and 12 methods. Most researchers have utilized these datasets independently to test the efficacy of their IDSs. The original dataset comprises 2,540,044 packets spread over four CSV files. Information types and their respective descriptions are included in Table 2.

3.3 Dataset splitting

A common approach for model validation is data splitting, which splits a given dataset into training and testing sets. The training data is then utilized to fit and evaluate statistics and ML models. It may test and compare the accuracy of several models' predictions without worrying about potential overfitting of the training set, provided it keeps a separate dataset for validation [19]. They can employ the aforementioned data-splitting strategies once they have defined a splitting ratio. A typical ratio is 80:20, which means that 80% of the data is utilized for training and 20% for testing [20]. In practice, alternative ratios such as 70:30, 60:40, and even 50:50 are used. There does not appear to be clear information on what ratio is ideal or best for a particular dataset. The 80:20 split is based on the well-known Pareto principle; however, this is merely a practice-based recommendation. Based on the theoretical or numerical study, there is no agreement on the optimal data-splitting ratio [21].

3.4 Dataset preprocessing

Data may be interpreted as the model algorithm performing a rapid study of the data's features. Data

Table 2: UNSW-NB15 Dataset categories [18]

Attack family	Description	No. of samples
Normal	Natural transaction data.	2,218,763
Analysis	Includes port scan, spam, and HTML file intrusion techniques.	2,677
Backdoor	A method for sneakily getting into a computer or its data by getting around a system's security.	2,329
DoS	The goal is to disable host services so users cannot access network resources.	16,353
Exploits	The attacker exploits a software or operating system security flaw.	44,525
Fuzzers	Sends a large amount of data to cause a computer program to crash.	24,246
Generic	A strategy is effective against all block ciphers regardless of the form of the block cipher.	215,481
Reconnaissance	The goal of this is to collect information.	13,987
Shellcode	Most of the time, a small piece of code is used to take advantage of a software flaw.	1,511
Worms	To get to other computers, the attacker copies itself. Most of the time, it spreads through a computer network.	174

preprocessing is the most important and critical stage for a DL algorithm to perform effectively in terms of generalization [22]. The training data grow exponentially in response to the input spatial dimension. It is predicted that preprocessing might take up to 50–80% of the time necessary for the whole classification process, highlighting its importance in model construction. The datasets need numerous preprocessing steps to contain undesirable elements such as missing, redundant, or infinite values that must be removed or changed to improve data quality for improved performance [23].

3.4.1 Dataset label encoder

Label encoding is a technique to preprocess categorical variables by assigning a unique integer to each label based on alphabetical ordering. These integers replace the variable in the same column. This is the method that was utilized in this study. The additional variables make the data more complex. For example, there are 135 variables in the "protocol" column, 13 in the "service" column, and 16 in the "status" column.

3.4.2 Apply min–max normalization

The min–max technique normalizes data for each feature by converting the minimum value to decimal numbers between 0 and 1. This ensures the data can be more easily understood and reduces training time. It is required when attributes have different scales [24]. According to equation (1),

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{1}$$

where $x_{\max}$ and $x_{\min}$ are the maximum and minimum feature values (x), resulting in output within the 0–1 range.

3.5 Classification algorithms

Five classification algorithms, i.e., CNN and NB models (Gaussian naïve Bayes [GNB], Bernoulli naïve Bayes [BNB], multinomial naïve Bayes [MNB], and complement naïve Bayes [CNB]), were employed to train the proposed model.

3.5.1 Data classification depending on CNN

CNN architecture includes convolution layers, pooling layers, and fully connected layers. A famous architecture is the recurrence of a stack of several convolution layers and a pooling layer, followed by one or more fully linked layers. Forward propagation is the method through which input data is converted into output data via these levels [25].

3.5.1.1 Convolution layer

The convolution layer, an essential component of CNN design, extracts features by combining linear and nonlinear operations, such as convolution and activation functions [26].

- **Convolution:** A linear procedure called convolution is utilized to extract features. A kernel is applied to the input, a tensor array of integers. A feature map, sometimes referred to as an output value at the corresponding position of the output tensor, is created by computing an element-wise product between each element of the kernel and the input tensor at each point of the tensor and adding the results.
- **Activation function:** The outputs of a linear operation, such as convolution, are then passed through a nonlinear activation function. The Leaky ReLU activation function was utilized in the proposed model. It is an effort to address the fading ReLU issue. When $x < 0$, a leaky ReLU will have a modest negative slope as opposed to zero (of 0.01 or so). The function computes this as shown in equation (2), where α is a small constant [27].

$$f(x) = 1(x < 0)(\alpha x) + 1(x \geq 0)(x). \tag{2}$$

3.5.1.2 Pooling layer

A pooling layer conducts a typical down-sampling procedure that reduces the in-plane dimensionality of the feature maps to introduce translation invariance to small shifts and distortions and to minimize the number of ensuing learnable parameters. It is crucial to note that none of the pooling layers has learnable parameters. However, filter size, stride, and padding are hyperparameters in pooling operations, just as they are in convolution operations.

- **Max pooling:** The most common sort of pooling procedure is max pooling. It accepts patches from the feature maps as input, outputs the most outstanding value in each patch, and discards all other values [28].

3.5.1.3 Fully connected layer

The final convolution or pooling layer's output feature maps are typically flattened or converted into a one-

dimensional (1D) array of numbers (or vectors) and connected to one or more dense layers, also known as fully connected layers, in which a learnable weight connects each input and output. A subset of fully connected layers maps the properties of convolution and down-sampling layers to the network's final outputs, such as the probabilities for each class in classification tasks. The number of output nodes in the final fully connected layer usually equals the number of classes [29].

- **SoftMax activation:** A different kind of LR called SoftMax Classifier may classify more than two classes. The output of the last layer may be transformed to its underlying probability distribution utilizing SoftMax. SoftMax has the advantage that the output probability can be between 0 and 1, and the sum of the probabilities is 1 [30].

3.5.1.4 Dropout layer

Dropout layers were used to keep from overfitting [29], which made the training last longer. It prevents overfitting by changing specific input units to 0 randomly throughout training. Those inputs not set to 0 are scaled by 1/(1 − rate) to keep the same total sum. So, there are dropout layers, and each one is meant to lower the chance of overfitting by making the neurons that come after it depend less on the neurons that came before it.

3.5.1.5 The proposed CNN-Dense model design

The proposed CNN-Dense model consists of 22 layers as follows:

- CNN layers (9).
- Max pooling layers (8).
- Dense layers (3).
- Flatten layer (1).
- Dropout layer (1).

Table 3 explains these layers in some detail.

Table 3: The proposed (CNN-Dense) layers and parameters settings

No.	Layer type	Filters	Size/stride	Activation function
1	Convolutional	16	3/1	Leaky ReLU
2	Max pooling	_	1/1	_
3	Convolutional	32	3/1	Leaky ReLU
4	Max pooling	_	1/1	_
5	Convolutional	64	3/1	Leaky ReLU
6	Max pooling	_	1/1	_
7	Convolutional	128	3/1	Leaky ReLU
8	Max pooling	_	1/1	_
9	Convolutional	128	3/1	Leaky ReLU
10	Max pooling	_	1/1	_
11	Dense	128	_	Linear
12	Convolutional	256	3/1	Leaky ReLU
13	Max pooling	_	1/1	_
14	Convolutional	512	3/1	Leaky ReLU
15	Max pooling	_	1/1	_
16	Convolutional	1,024	3/1	Leaky ReLU
17	Max pooling	_	1/1	—
18	Dense	1,024	_	Linear
19	Convolutional	64	3/1	Leaky ReLU
20	Flatten	_	_	_
21	Dropout	_	0.1	_
22	Dense	_	_	SoftMax

3.5.2 NB classifier

The NB classifier applies the Bayes theorem and operates on the probabilistic premise that features are independent and equally weighted [31–36]. One of the NB challenges is the zero frequency or probability scenario, in which the model cannot forecast if it has not seen a specific category in the training dataset yet does so in the test dataset when it encounters a novel and previously unknown input variable. Laplace estimates and other smoothing techniques can be used to prevent this undesired situation [32]. Because it is straightforward to implement, computationally quick, and resilient, it has been widely used in text classification and other classification domains, with several changes to the traditional NB [33]. NB is one of the simplest Bayesian network methods, and when combined with kernel density estimation, it could be more accurate [34].

4 Evaluation metrics

The confusion matrix, considering the calculated predicted class vs the actual class variables, defines various performance metrics [35,36].

- **True positive (TP):** The number of harmful codes that have been accurately discovered.
- **True negative (TN):** The number of innocuous codes successfully identified.
- **False positive (FP):** The number of times a detector incorrectly identifies a benign file as malware.
- **False negative (FN):** The number of malicious codes detected by a detector incorrectly since the virus is new and no signature is yet accessible.
- **Accuracy:** It indicates the accuracy or proximity of the estimated value to the actual value of the model,

implying that a part of the total samples is properly classified. The model's accuracy is calculated using the following formula:

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

- **Precision:** It indicates the percentage of relevant occurrences that are genuinely positive among the selected instances. To calculate precision, use the following formula:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (4)$$

- **Recall or true positive rate (TPR):** It computes the percentage of true positives that are accurately detected. The recall formula is as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (5)$$

- ***F*1-score:** The harmonic mean of accuracy and memory is interpreted as the *F*1-score, which combines the weighted average of precision and recall. *F*1-score is calculated using the following formula:

$$F1_{\text{Score}} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}. \quad (6)$$

5 Experiment and result analysis

The experimental setup was created utilizing the methods described in Figure 2. The UNSW-NB15 Dataset was divided first in this study. Then, data standardization is utilized to rescale the dataset's data values. Finally, five classification algorithms, including CNN-Dense and four NB models, are employed to differentiate between attack groups and regular traffic.

5.1 Performance analysis of the proposed CNN-Dense classification model

In the first method, the proposed CNN-Dense model was used, as shown in Table 4.

From Table 4, it is clear that the results of the method were perfect, the accuracy of the detection was very high, and the best value reached 99.8%. The reason for this is that the techniques used, whether in the preprocessing of the data or the proposed model structure of a one-dimensional convolutional neural network (1D-CNN), have significantly reduced the computational complexity and speed of intrusion detection.

Table 4: Results of the proposed CNN-Dense model

Accuracy (%)	Precision (%)	Recall (%)	*F*1-score (%)
99.8	99.89	99.76	99.8

5.2 Performance analysis of NB classification models

In the second method, the NB models were used, as shown in Table 5.

5.3 Comparison between DL and ML models results

The results of the first suggested DL model are better than those of ML because it can improve results automatically and without human intervention through a process called backpropagation. It can also use considerable datasets in real time, which may contain many different kinds of unstructured data. Figure 3 shows the chart of this comparison.

5.4 Comparison with previous studies

The comparison of results with the related studies mentioned in Section 2 is explained in Table 6 and Figure 4.

The results in Table 6 and Figure 4 clearly show that the proposed method, based on a 1D-CNN, gave the best intrusion detection accuracy compared to our second method or related studies.

Table 5: Results of the NB models

Technique	Accuracy (%)	Precision (%)	Recall (%)	*F*1-score (%)
GNB	83	84	81	82
BNB	69	70	67	67
MNB	80	74	77	75
CNB	80	81	78	78

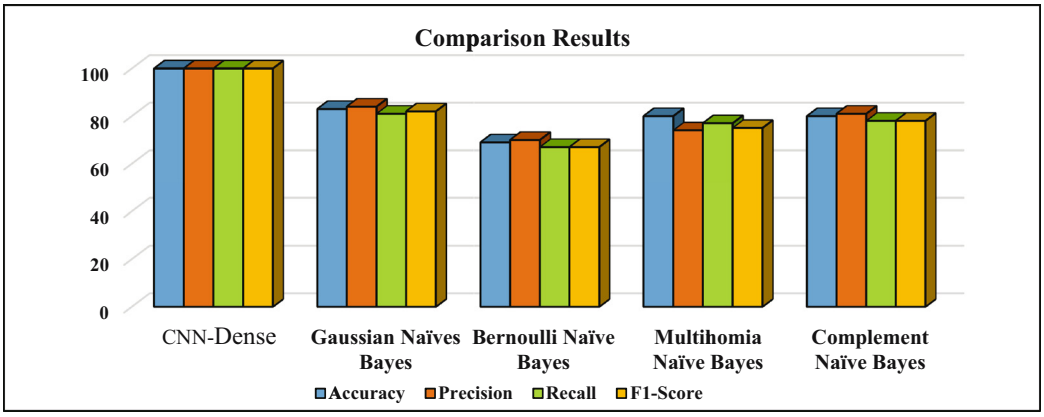

Figure 3: ML and DL experimental results comparison.

Table 6: Comparison results with related studies

Ref.	Technique	Dataset	Accuracy (%)	Precision (%)	Recall (%)	*F*-score (%)
[10]	DT	UNSW-NB15	90.74	93	89.78	89.1
[11]	ANN	UNSW-NB15	77.51	77.51	77.51	77.51
[12]	DT	UNSW-NB15	89.76	90	89.23	89.12
[13]	RF	UNSW-NB15	95.9	96.9	95.9	95.9
[14]	DNN	UNSW-NB15	84.98	85.1	84.8	84.98
[15]	CNN	UNSW-NB15	93.5	93	93.56	93.6
[16]	CNN	UNSW-NB15	76.3	90.4	76.1	78.2
[17]	CNN	UNSW-NB15	92.10	93	92.1	92.1
Our methods	(CNN-Dense)	UNSW-NB15	99.8	99.89	99.76	99.8
	GNB	UNSW-NB15	83	84	81	82
	BNB	UNSW-NB15	69	70	67	67
	MNB	UNSW-NB15	80	74	77	75
	CNB	UNSW-NB15	80	81	78	78

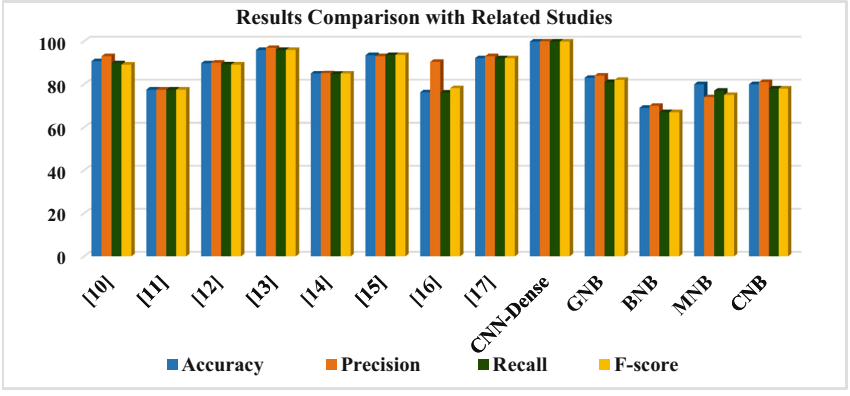

Figure 4: Comparison results of our proposed models with the previous studies.

6 Conclusions

This study presents a framework for detecting network intrusions. The suggested framework's performance was analyzed and assessed using the UNSW-NB15 Dataset. Preprocessing steps such as label encoders and normalization datasets are important stages before performing DL algorithms. Its purpose is to initialize the data and reduce the complexity of the calculations in the algorithm. The results demonstrated significantly improved accuracy, particularly in the first hybrid technique (CNN-Dense). Based on the evaluation findings, it can be concluded that the proposed classifier outperformed the ML and DL models on the UNSW-NB15 Dataset in terms of accuracy, precision, recall, and *F*1-score metrics.

In the future, reduction techniques can be used to reduce the features, or the deep model from one layer can be used to extract features faster and input the extracted features to ML for classification, as well as using another dataset with the proposed model and evaluating the classified data to detect network intrusion.

Conflict of interest: The authors declare that they have no conflict of interest.

Data availability statement: Most datasets generated and analyzed in this study are comprised in this submitted manuscript. The other datasets are available on reasonable request from the corresponding author with the attached Information.

References

[1] Wu M, Moon Y. Intrusion detection system for cyber manufacturing system. J Manuf Sci Eng. 2019 Jan;141(3):031007.

[2] Mujeeb Ahmed C, Umer MA, Binte Liyakkathali BS, Jilani MT, Zhou J. Machine learning for CPS security: Applications, challenges, and recommendations. Machine intelligence and big data analytics for cybersecurity applications. Cham: Springer; 2021. p. 397–421.

[3] Prasad R, Rohokale V. Artificial intelligence and machine learning in cyber security, cyber security: The lifeline of information and communication technology. New York, NY: Springer; 2020. p. 231–47.

[4] Alheeti K, Alsukayti I, Alreshoodi M. Intelligent botnet detection approach in modern applications. Int J Interact Mob Technol (IJIM). 2021;15(16):113–26.

[5] Obeidat I, Hamadneh N, Alkasassbeh M, Almseidin M, AlZubi MI. Intensive preprocessing of KDD Cup 99 for network intrusion classification using machine learning techniques. Int J Interact Mob Technol (IJIM). 2019;13(1):70.

[6] Mishra P, Varadharajan V, Tupakula U, Pilli ES. A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE Commun Surv Tutor. 2019;21(1):686–728.

[7] Moustafa N, Slay J. The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Inf Secur J A Glob Perspect. 2018;25:18–31.

[8] Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSP. 2018;1:108–16.

[9] Umer MA, Junejo KN, Jilani MT, Mathur AP. Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations. Int J Crit Infrastruct Prot. 2022;38:100516. arXiv:2202.11917v1 [cs.CR] 24 Feb 2022.

[10] Kumar V, Sinha D, Das AK, Pandey SC, Goswami RT. An integrated rule based intrusion detection system: Analysis on UNSW-NB15 data set and the real time online dataset. Clust Comput. 2020;23:1–22.

[11] Kasongo SM, Sun Y. Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. J Big Data. 2020;7(1):38367.

[12] Kumar V, Das AK, Sinha D. UIDS: A unified intrusion detection system for IoT environment. Evolut Intell. 2021;14(1):47–59.

[13] Shushlevska M, Efnusheva D, Jakimovski G, Todorov Z. Anomaly detection with various machine learning classification techniques over UNSW-NB15 dataset. 10th International Conference on Applied Innovations in IT, (ICAIIT); March 2022. p. 21–7.

[14] Wu P, Guo H. LuNET: a deep neural network for network intrusion detection. In 2019 IEEE symposium series on computational intelligence (SSCI); 2019. pp. 617–624.

[15] Mahalakshmi GN, Uma E, Aroosiya M, Vinitha M. Intrusion detection system using convolutional neural network on UNSW NB15 dataset. Adv Parallel Comput Technol Appl. 2021;40:1–8.

[16] Singh NB, Singh MM, Sarkar A, Mandal JK. A novel wide & deep transfer learning stacked GRU framework for network intrusion detection. J Inf Secur Appl. 2021;61:102899.

[17] Almarshdi R, Nassef L, Fadel E, Alowidi N. Hybrid deep learning based attack detection for imbalanced data classification. Intell Autom Soft Comput. 2022;35(1):297–320.

[18] Rashid OF. DNA encoding for misuse intrusion detection system based on UNSWNB15 data set. Iraqi J Sci. 2020 Dec;61(12):3408–16. doi: 10.24996/ijs.2020.61.12.29.

[19] Nurhopipah A, Hasanah U. Dataset splitting techniques comparison for face classification on CCTV images. Indones J Comput Cybern Syst. October 2020;14(4):341–52.

[20] Nguyen QH, Ly HB, Ho LS, Al-Ansari N, Le HV, Tran VQ, et al. Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. Math Probl Eng. 2021;2021:1–15.

[21] Awwalu J, Nonyelum O. On holdout and cross-validation: A comparison between neural network and support vector machine. Int J Trend Res Dev 6(2):235–9.

[22] Huang F. Data processing. In: Schintler L, McNeely C, editors. Encyclopedia of big data. Cham: Springer; 2019.

[23] Abdulrahman AA, Ibrahem MK. Intrusion detection system using data stream classification. Iraqi J Sci. Jan. 2021;62(1):319–28. doi: 10.24996/ijs.2021.62.1.30.

[24] Raju VG, Lakshmi KP, Jain VM, Kalidindi A, Padma V. Study the influence of normalization/transformation process on the accuracy of supervised classification. In 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT). IEEE; 2020. p. 729–35.

[25] Chen MC, Ball RL, Yang L, Moradzadeh N, Chapman BE, Larson DB, et al. Deep learning to classify radiology free-text reports. Radiology. 2018;286:845–52.

[26] Bezdan T, Džakula N. Convolutional neural network layers and architectures. International Scientific Conference On Information Technology and Data Related Research; 2019.
[27] Sultana F, Sufian A, Dutta P. Advancements in image classification using convolutional neural network. In 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN). Kolkata, India: IEEE; 2018. p. 122–9.
[28] Thirimanne SP, Jayawardana L, Yasakethu L, Liyanaarachchi P, Hewage C. Deep neural network based real-Time intrusion detection system. SN Comput Sci. 2022;3(145):145.
[29] Yamashita R, Nishio M, Do R, Togashi K. Convolutional neural networks: An overview and application in radiology. Insights Imaging. 2018;9:611–29.
[30] Ren S, He K, Girshick R, Sun J. Faster RCNN: Towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2017;39(6):1137–49.
[31] Granik M, Mesyura V. Fake news detection using naïve Bayes classifier. IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON). Kie; 2017. p. 900–3.
[32] Xu S. Bayesian naïve Bayes classifiers to text classification. J Inf Sci. 2018;44(1):48–59.
[33] Sasongko TB, Arifin O, Al Fatta H. Optimization of hyper parameter band-width on naïve Bayes kernel density estimation for the breast cancer classification. In 2019 International Conference on Information and Communications Technology (ICOIACT). IEEE; 2019. p. 226–31.
[34] Anand MV, KiranBala B, Srividhya SR, C. K, Younus M, Rahman MH. Gaussian naïve Bayes algorithm: A reliable technique involved in the assortment of the segregation in cancer. Hindawi. Mob Inf Syst. 2022;2022:1–7.
[35] Jabbar AF, Mohammed IJ. BotDetectorFW: An optimized botnet detection framework based on five features-distance measures supported by comparisons of four machine learning classifiers using CICIDS2017 dataset. Indones J Electr Eng Comput Sci. Jan. 2021;21(1):377–90. doi: 10.11591/ijeecs.v21.i1.pp377-390.
[36] Mahmood RAR, Abdi A, Hussin M. Performance evaluation of intrusion detection system using selected features and machine learning classifiers. Baghdad Sci J. 2021;18(2):884–98.