ISSN 2723-6471

Assimilate Grid Search and ANOVA Algorithms into KNN to Enhance Network Intrusion Detection Systems

Mohammad A. Alsharaiah[1], Mohammed Amin Almaiah[2,*], Rami Shehab[3,*], Tayseer Alkhdour[4], Rommel AlAli[5], Fares Alsmadi[6]

[1,2,6]*The University of Jordan, King Abdullah II School of Information Technology, Department of Information Technology. Amman, Jordan*

[3,4] *Vice-Presidency for Postgraduate Studies and Scientific Research, King Faisal University, Al-Ahsa 31982, Saudi Arabia*

[5]*The National Research Center for Giftedness and Creativity, King Faisal University, Al Hofuf 31982, Saudi Arabia*

(Received: January 15, 2025; Revised: March 12, 2025; Accepted: April 10, 2025; Available online: June 2, 2025)

Abstract

The recent progress of operational network intrusion detection systems (NIDS) has become increasingly essential. Herein, a fruitful attempt to introduce an innovative NIDS methodology that integrates the grid search optimization algorithm and ANOVA techniques with the K nearest neighbor (KNN) algorithm to analyze both spatial and temporal characteristics of data for network traffic. We employ the UNSW-NB15 benchmark dataset, which presents various patterns and a notable imbalance between the training and testing data, with 257674 samples. Therefore, the Synthetic Minority Oversampling Technique has been used since this method is effective in handling imbalanced datasets. Further, to handle the overfitting issue the K folds cross-validation method has been applied. The feature sets within the dataset are meticulously selected using ANOVA mechanisms. Subsequently, the KNN classifier is fine-tuned through hyperparameter tuning using the grid search algorithm. This tuning process includes adjusting the number of K neighbors and evaluating various distance metrics such as 'euclidean', 'manhattan', and 'minkowski'. Herein, all attack types in the dataset were labeled as either 1 for abnormal instances or 0 for normal instances. Our model excels in binary classification by harnessing the strengths of these integrated techniques. By conducting extensive experiments and benchmarking against cutting-edge machine learning and deep learning models, the effectiveness and advantages of our proposed approach are thoroughly demonstrated. Achieving an impressive performance of 99.1%. Also, several performance metrics have been applied to assess the proposed model's efficiency.

Keywords: UNSW-NB15, Classification, ANOVA, Grid Search, KNN

1. Introduction

The extensive implementation of internet technologies, especially with the incorporation of cloud facilities, has resulted in an important rise in intrusion occasions. Prominent platforms such as Google and Amazon, which contain countless servers and offer services to numerous organizations, have become prime objects for malicious actions. As a result, societies are incurring increasing charges to implement security measures like firewalls to safeguard their data and maintain continuous service. Failure to identify and respond to intrusions can lead to severe repercussions for an organization's character and the reliability of its data [1].

Network Intrusion Detection Systems (NIDS) are vital for defending networks against malicious actions [2]. This type of routine, which must be implemented in either hardware or software, is built to monitor and notice illegal or malicious network traffic. NIDS works by passively observing network traffic, examining it for unusual patterns, and matching these patterns with a database of recognized attack signatures [3].

Thorough traffic surveillance is guaranteed by strategically positioning NIDS at crucial locations within a network, such as on network channels or particular hardware. NIDS thoroughly examines all network traffic inside a subnet and matches it to a pre-established database of attack signs. When anomalous activities or an attack is detected, NIDS generates alerts to notify network administrators, prompting them to examine and take suitable movements [4]. The

***Corresponding author: Mohammed Amin Almaiah (m_almaiah@ju.edu.jo) and Rami Shehab (Rtshehab@kfu.edu.sa)**
DOI: https://doi.org/10.47738/jads.v6i2.604

 ISSN 2723-6471

effectiveness of NIDS lies in their capability to catch and reply to suspicious patterns crossways various hosts, facilitating the early recognition and avoidance of attacks earlier they can influence their goals. As a result, NIDS serves as a vital security measure, constantly monitoring network packets and quickly warning administrators of possible threats.

Conversely, it is important to recognize the challenges and constraints encountered by NIDS [5]. These comprise the essential for incessant information to attack signature databases to keep ahead of emerging threats, the impending untrue positives or untrue negatives when detecting intrusions, and the measurement issues NIDS faces when managing large volumes of network traffic. Overcoming these obstacles calls for the creation of advanced detection methods that integrate machine learning (ML) and deep learning (DL) algorithms to improve the precision and performance of IDS [6].

The arrangement of this study is prepared as follows: Section 1 delivers an introduction to the study, forming its context and objectives. Section 2 compromises a comprehensive review of the related literature, stress-standing studies, and their relevance. Section 3 presents a complete explanation of the primary work undertaken. Section 4 conveys an in-depth explanation of the dataset employed in the study. Section 5 expands on the data preprocessing methods employed to arrange the data for examination. Section 6 conducts a thorough investigation of the outcomes and occupies a serious argument of their implications. Finally, Section 7 accomplishes the research by summarizing the conclusions and deliberating their broader significance.

2. Literature Review

NIDS are dynamic for protective networks against malicious actions and must be set up in organization networks as represented in figure 1. Therefore, various methods have been implemented (NIDS) to categorize network data packets as either normal or abnormal. For example, ML-based NIDS uses specific frameworks like kernel machines and ensemble methods for classification [7], [8]. A frequently used classification technique is the support vector machine (SVM), which leverages kernel machines and the Gaussian kernel [9]. The kernel function enables the SVM to handle nonlinear datasets by representing the data in a further-dimensional space, thus making it linearly separable. Additionally, ensemble classifiers combine various non-powerful classifiers into a single strong model to mitigate overfitting during training. Techniques like random forest [10], [11] and adaptive boosting [12], [13] exemplify this robust classification approach. Machine learning techniques emphasize understanding the importance of several features. Methods for feature obtainability also utilize dimensionality reduction procedures to recognize the most ideal relationships amongst dataset features. Additionally, these techniques aim to predict the finest outcomes with the optimum time steps.

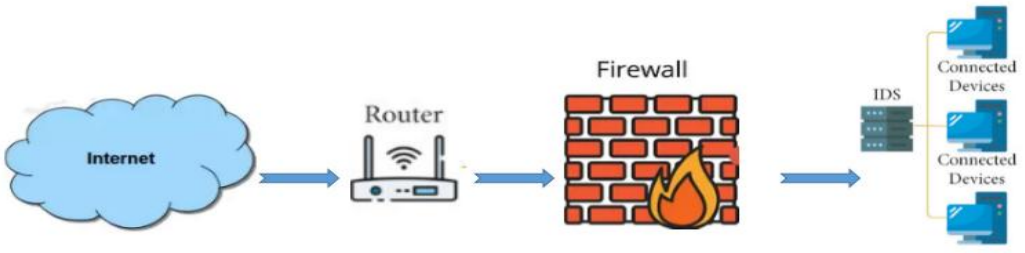

Figure 1. NIDS Topology

Recently, hierarchical models have been utilized to detect mutually spatial and temporal patterns in network traffic data. For example, predictive models designed for network time series data have gained significant popularity. This kind of dataset exhibits nonlinear characteristics due to the fluctuating nature of several data points over time, causing irregular fluctuations. ML algorithms such as naïve Bayes and support vector machines have been applied to develop NIDS [14]. However, these arithmetical methods do not account for mutual relationships among data points. Despite their potential, these models are still under research for practical applications due to their high false positive rates [15]. Specifically, the model must be trained on a dataset with features that characterize typical network behavior. Generally, labels are assigned as 0 for normal behavior and 1 for attacks. Although certain datasets categorize attacks into multiple

subtypes, this method can be time-intensive and less precise. The literature presents several ML models for classification. The Bayesian model [16], [17], for example, uses Bayes' theorem to train data and create a classification model. Hierarchical Intrusion Detection Employing ML and Knowledge Model. Consequently, adopting digital learning platforms at the higher education level offers the potential to transform learning and education by enhancing teachers' digital pedagogical skills and providing students with more equitable and sustainable learning opportunities. Additionally, various hybrid models have been introduced, including [18], which suggests a combined feature selection approach that integrates Information Gain, Random Forest Importance, and Recursive Feature Elimination to improve the performance of a Multilayer Perceptron (MLP). [19] introduces a dataset-based approach for designing an ensemble classifier specifically for the UNSW-NB15 dataset, tackling issues like class imbalance and feature overlap.

This study presents a unified (IDS) that combines ML algorithms with knowledge-based approaches to effectively detect and classify network intrusions. The IDS integrates the grid search optimization algorithm and ANOVA techniques with the K-nearest neighbour (KNN) algorithm to examine the spatial and temporal characteristics of data traffic inside the network. For evaluation, the UNSW-NB15 benchmark dataset is utilized [20], proposing a varied series of patterns and an inherent imbalance between training and testing sets. The selection of features from the dataset is conducted rigorously through ANOVA-based mechanisms [21]. Subsequently, the KNN classifier undergoes hyperparameters optimization via grid search. The proposed IDS's performance is further evaluated using the KDD 99 dataset, enabling a benchmark comparison with other established methodologies [22].

3. Methodology

The proposed model introduced an innovative methodology that integrates Grid search optimization and ANOVA with the KNN algorithm to analyze spatial and temporal characteristics of the imbalanced dataset. The imbalanced dataset was treated by using the synthetic Minatory Oversampling technique (SMOTE). The Framework graph outlines the proposed study methodology as shown in figure 2. Mainly, the graph breaks down the model's workflow and highlights the integration of key components.

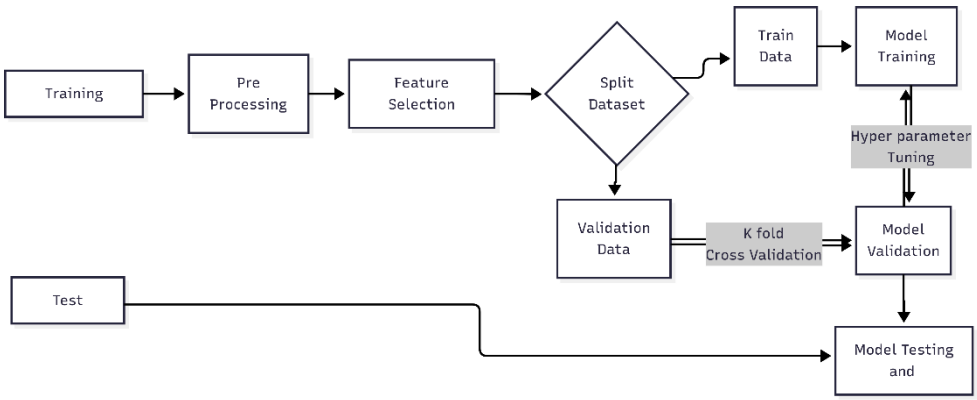

Figure 2. Proposed model methodology

The process begins with data collection using the NB15 dataset, which serves as the basis for both training and testing the model [23]. Following this, data exploration is conducted to gain a deeper understanding of the features and patterns within the dataset. This includes steps such as data cleaning, normalization, and the identification of trends or relationships. Subsequently, feature selection is carried out using the ANOVA method to pinpoint the most impactful features. This technique not only reduces dimensionality by focusing on features with statistically significant contributions but also enhances model accuracy by eliminating noise and irrelevant attributes. Moreover, it helps in distinguishing between normal and malicious network traffic.

For model selection and optimization, the KNN algorithm is chosen due to its simplicity and effectiveness in binary classification tasks. To further refine the model, grid search optimization is applied to fine-tune key hyperparameters. This systematic approach evaluates combinations of parameter values, including the number of neighbors (k), the distance metric (such as Euclidean or Manhattan), and the weighting scheme (uniform or distance-based), to identify the optimal configuration. The training phase involves applying k-fold cross-validation to the selected features, which

 ISSN 2723-6471

helps mitigate overfitting. The model is then tested on a separate subset of the NB15 dataset to assess its generalization performance. The classification task involves two main goals: identifying whether an attack is present (binary classification) and predicting the specific category of the detected attack. Finally, model assessment is conducted using evaluation metrics such as accuracy, precision, recall, and the confusion matrix, providing a comprehensive understanding of the model's performance.

3.1. Data Set

This analysis utilizes the standard dataset UNSW-NB15, developed by Mustafa and Slay [19]. They analyzed typical network traffic patterns and categorized modern attack procedures into nine different types. The UNSW-NB15 dataset comprehends a broad range of mutually actual and imitation network traffic attacks. Moreover, features for the dataset were generated through a combination of conventional and novel approaches. The UNSW-NB15 dataset contains 49 attributes, which can be classified into numerous sets. We split the dataset into training and testing segments. Table 1 highlights specific features of the dataset, including data kinds and traffic attribute classes [24].

Table 1. The Features in UNSW_NB15 regarding data types and classes

No.	Group	Name	Data Type	No.	Group	Name	Data Type
1		Dstip	Nominal	27		Syncak	Float
2		Sport	Integer	28		Djit	Float
3	Flow	Proto	Nominal	29		ackdat	Float
4		Dsport	Integer	30		Ltime	Timestamp
5		scrip	Nominal	31	Time	Sintpkt	Float
6		Service	Nominal	32		Dintpkt	Float
7		Dur	Float	33		Tcprit	Float
8		dttl	Integer	34		Sjit	Float
9	Basic	Gloss	Integer	35		stime	Timestamp
10		sttl	Integer	36		Ct_ftp_cmd	Integer
11		sload	Float	37		Is_ftp_login	Binary
12		sloss	Integer	38	General Purpose	Ct_flw_http_mthd	Integer
14		state	Nominal	39		Ct_state_til	Integer
15		sbytes	Integer	40		Is_sm_ips_ports	Integer
16	General Purpose	Diaod	Float	41		Ct_dst_sport_ltm	Integer
17		Spkts	Integer	42		Ct_srv_ltm	Integer
18		Dpkts	Integer	43		Class	Integer
19		Swin	Integer	44		Ct_src_ltm	Integer
20		Dmeansz	Integer	45	Connection	Ct_src_doprt_ltm	Integer
21		Stcpb	Integer	46		Attack_cat	Nominal
22	Content	Dtcpb	Integer	47		Ct_dst_src_ltm	Integer
23		Smeanz	Integer	48		Ct_dst_ltm	Integer
24		dwin	Integer	49		Ct_srv_src	Integer
25		res_bdy_len	Integer				
26		trans_depth	Integer				

3.2. Data Set Pre-Processing

The dataset required a preprocessing phase, which involved several phases, such as data balancing, normalization, cleansing, discount, transformation, and feature selection. These stages are crucial as they directly touch the power of the classifier model [25]. The UNSW-NB15 has an inherent imbalance between training and testing data. Addressing this issue, we already have applied class balancing techniques over the model training process to mitigate the impact of this imbalanced data set. Specifically, SMOTE has been used since this method is effective in handling imbalanced datasets [3]. Herein, the inputs are carefully selected from the UNSW-NB15 dataset for the suggested model, ensuring that the essential data preprocessing phases are applied. For example, the chosen training sets from UNSW-NB15 are free from redundant and duplicate records. Categorical variables, such as protocol and service types, were present in the NB15 dataset [26]. These variables were transformed into numerical values to make them suitable for machine learning algorithms, which typically require numerical inputs. We employed one-hot encoding for these categorical features, which involves creating a new binary column for each possible category. This transformation ensures that the

 ISSN 2723-6471

model can interpret the categorical data without implying any inherent ordering or ranking, as would be the case with label encoding [27].

Data normalization was then performed by scaling the amount of every feature within a comparative variety, typically (0, 1), to prevent the dataset from being biased toward features with naturally higher values [28]. This step was carried out using the method outlined in Equation 1. Next, categorical data were converted into numeric values, and all attack types in the dataset were labeled as either 1 for abnormal instances or 0 for normal instances. This step simplifies the classification task and focuses on detecting malicious activities versus benign activities subsequently, only the regular data trials were extracted from the training sets [29], [30]. This extraction was necessary because the proposed model is designed to be applied during the early phases of a network's lifecycle, where it can initially learn from normal traffic. In the later stages, during the testing phase, the model should be capable of distinguishing between regular and attack traffic. Last of all, to confirm optimal training speed, the normal traffic instances obtained were gathered during the training phase to reduce the number of training registers.

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

One-hot encoding allowed the model to effectively process categorical data, while normalization ensured that all features were scaled uniformly, enabling the KNN model to make precise distance-based decisions. These preprocessing techniques were crucial in minimizing model bias and enhancing overall classification performance. Furthermore, classification accuracy was chosen as the basis for assessment in terms of the performance metric used to evaluate the various procedures compared with the proposed model [4]. The formula for accuracy, recall, precision, and F-Measurement are provided in Eq. 2,3,4,5 respectively.

$$Accuracy = \frac{TP+TN}{TN+TP+FN+FP} \quad (2)$$

$$\text{Recall} = \text{True Positive} / (\text{True Positive} + \text{False Negative}) \quad (3)$$

$$\text{Precision} = \text{True Positive} / (\text{True Positive} + \text{False positive}) \quad (4)$$

$$\text{F-Measure} = 2 * \text{precision} * \text{Recall} / (\text{precision} + \text{Recall}) \quad (5)$$

However, accuracy alone is not sufficient for a thorough evaluation, especially for classification models. While accuracy is commonly used, it can be misleading in imbalanced datasets, such as those found in NIDS, as it may mask poor detection of minority classes, potentially leaving malicious activities undetected. Metrics like precision, recall, and the F-measure offer a more comprehensive evaluation. Precision helps minimize false positives, recall reduces false negatives, and the F-measure provides a balance between both [31], [32].

As a result, we have included additional evaluation metrics, such as the error matrix (or confusion matrix). This matrix provides a clear visualization of the algorithm's performance, where each row represents the actual class and each column the predicted class, as shown in figure 3. True Positives (TP) indicate the number of correctly identified intrusions, with a high TP count demonstrating the model's effectiveness in detecting malicious activities [33], [34]. True Negatives (TN) reflect the number of correctly identified benign activities, ensuring that normal traffic is not mistakenly flagged as malicious and minimizing unnecessary alerts [35], [36]. False Positives (FP) correspond to benign activities incorrectly classified as intrusions, which can overload security personnel with false alarms and contribute to alert fatigue [37], [38]. False Negatives (FN) represent actual intrusions that the system fails to detect, creating a serious security risk as malicious activities may go unnoticed [39], [40].

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

Figure 3. Confusion Matrix metric

 ISSN 2723-6471

4. Findings and Discussions

4.1. Data examination and simulation Trials

This study conducted extensive trials on the proposed model for binary classification using the UNSW-NB15 dataset. The model was trained specifically on a Denial of Service (DoS) subset, detailed in table 1, with 82,323 samples used for training and 175,341 for testing. To evaluate the model's efficiency, several validation methods are available; cross-validation methods are commonly employed to gauge prediction accuracy. In this study, at the beginning of the initial phase, we used K-fold cross-validation with K set to 1, which resulted in a straightforward train/test split to evaluate the model's efficiency in binary classification with minimal computational overhead. This initial setup allowed for rapid prototyping and early detection of potential issues such as increasing the risk of overfitting. We recognize the usage of K=1 increased the risk of overfitting due to the lack of diverse validation sets. Therefore, to address this, the progression to K=10 cross-validation in the subsequent experiments was made. This adjustment provided a more comprehensive evaluation of model generalization capabilities by ensuring that each subset of the data was used for both training and validation [41], [42].

However, for deep learning-based NIDS, there are additional techniques to prevent overfitting, especially as these models are often more prone to overfitting due to their complexity [43], [44]. For instance, early stopping is a regularization technique that involves monitoring the model's performance on the validation set during training. If the validation performance starts to degrade while the training performance continues to improve, training is halted early to prevent overfitting. This helps in finding the optimal model before it starts to memorize the training data.

Herein, the proposed (NIDS) framework is implemented via Scikitlearn library and Python programming languages. They were utilized to train the model for Binary classification. Concerning the task of categorizing data. The implementation of the KNN model incorporated several strategic enhancements aimed at optimizing performance and achieving accurate predictions [45], [46]. To begin with, feature scaling was applied using the StandardScaler function to standardize all features to a uniform scale. This step is crucial for distance-based algorithms like KNN, as it prevents features with larger value ranges from exerting undue influence on the model’s predictions [47], [48].

Looking further to strengthen model robustness, Feature selection was undertaken to explore the potential benefits of adding, removing, or transforming features based on domain-specific knowledge. The goal of these adjustments is to advance the model's generality capabilities by aligning input features more accurately with the underlying patterns in the data. Initially, we assessed the relationships between features through correlation analysis as shown in figure 4. Also, an Analysis of Variance (ANOVA) to evaluate the significance of these relationships. ANOVA is computationally efficient when compared to more complex feature selection methods such as Recursive Feature Elimination (RFE). It evaluates features independently, without involving iterative processes or requiring the training of machine learning models, making it suitable for large datasets0. ANOVA can help in avoiding overfitting since it focuses on selecting features that are truly relevant to the target variable, as consequent it reduces the risk of including irrelevant features, which can lead to overfitting in downstream models. In addition, compatibility with Other Techniques. ANOVA works well as a preprocessing step for machine learning pipelines such as including KNN [3].

ANOVA, a statistical method, is employed to regulate whether there are substantial variances in the means among various categories. Within feature selection, ANOVA is particularly useful for assessing whether certain features show important variation across categories or groups in the data. This insight is valuable for understanding feature importance, aiding in the selection of features that are most likely to enhance the predictive model's capability to differentiate amongst classes or groups. Figure 4 represents the feature correlation analysis.

 ISSN 2723-6471

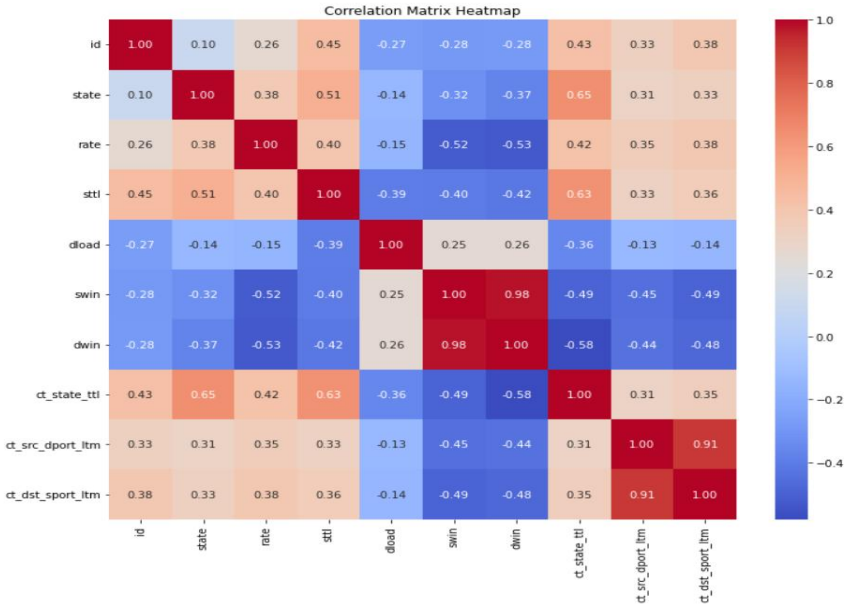

Figure 4. Feature correlation analysis.

Furthermore, hyperparameter optimization was carried out using GridSearchCV to determine the best KNN settings. By exhaustively searching through a specified parameter grid, GridSearchCV identifies the combination of parameters that maximizes model accuracy. The KNN model was then trained with these optimized parameters, ensuring it was fine-tuned to perform optimally with the dataset. In addition, hyperparameter optimization was conducted using GridSearchCV to determine the optimal settings for the KNN model. Table 2 represents a detailed description of the grid of hyperparameters tested. Specifically: For K in KNN, we tested several values such as Neighbors values, several distance metrics, and Additional hyperparameters.

Table 2. hyperparameters evaluation set for optimization to use in grid search

Neighbors' values	distance metrics	Additional hyperparameters
[11 ,9 ,7 ,5 ,3]	Euclidean, minkowski and Manhattan	uniform and distance

Through an exhaustive search across a defined parameter grid, GridSearchCV identified the parameter combinations that maximized model accuracy. The KNN model was subsequently trained with these optimized parameters, ensuring it was fine-tuned to deliver peak performance on the dataset. On accomplishment trials, it is possible to be assured that the detectives take finalized wide trials on the KNN model via attempting varied hyper-parameters, as shown in figure 5.

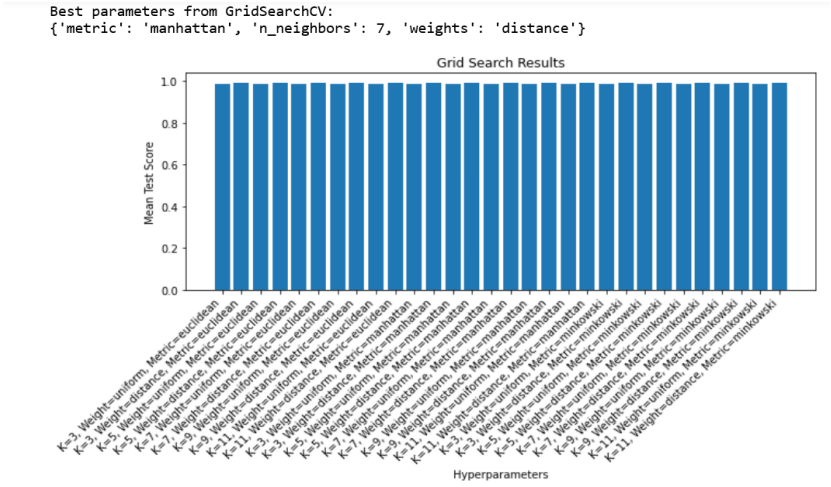

Figure 5. Grid search hypermeters optimization result

 ISSN 2723-6471

4.2. Outcomes Exploration

Many ML models that perform binary classification on datasets like UNSW-NB15 rely on feature engineering due to the high dimensionality of the data [31], [32],[33], [34], [35]. For example, [6] applied a filter-based feature reduction technique, while [7] employed a wrapper-based feature extraction approach. Additionally, [8] developed a hybrid model combining a Genetic Algorithm (GA) with logistic regression, tested on both the UNSW-NB15 and KDDCup99 datasets. The results demonstrated that this model achieved an overall accuracy of 81.42% when using the full feature set. [9] introduced a classifier based on the Least Squares Support Vector Machine (LS-SVM), which attained an accuracy of 78.86%. Meanwhile, [10] The authors performed a thorough analysis of the dataset using the Weka instrument, applying numerous algorithms including the feature assessor, Ranker method, Greedy Stepwise, and Information Gain. They evaluated the performance of each subset through the Kappa Statistic, and the findings indicated that the tRF classifier emerged as the most effective approach, achieving an accuracy rate of 75.66%.

Another IDS system was developed utilizing the UNSW-NB15 dataset by Kappa Statistic, where the authors incorporated the Information Gain method into their model, alongside a combined rule-based approach that involved manifold tree classifiers [36], [37], [38], [39], [40]. The outcomes from their IDS model showed an accuracy of 57.01% [11]. Though, they suggested that the model could be improved by incorporating substitute machine learning techniques, rather than focusing solely on tree-based approaches [41], [42], [43], [44], [45]. Maajid and Nalina [12] proposed an IDS system established on ML, such as the Random Forest (RF) algorithm. The results indicated that the RF algorithm achieved the best performance, with a correctness of 75.56%. Gao et al. [13] introduced a mixture IDS system that utilized an Advanced Principal Component Analysis (APCA) method combined with an enhanced version of the Extreme Learning Machine (IELM). This model was trained and tested on the dataset, and the results revealed that the IELM-APCA approach attained an accuracy of 70.51%. Additionally, Kaiyuan et al. [14] proposed an integrated NIDS framework that merges CNN with Bidirectional Long-Short Term Memory (Bi-LSTM). This CNN-Bi-LSTM model was taught and assessed by the aforementioned dataset. The results revealed that the model of CNN-Bi-LSTM realized a correctness of 77.16%.

In this research, the proposed model conducted the cross-validation into the GridSearchCV process to guarantee a consistent evaluation of the model. This approach evaluates the model's efficiency through multiple data partitions, decreasing the risk of overfitting and giving a further accurate estimate of its predictive capabilities. These combined strategies fortify the KNN model, supporting a rigorous approach to binary classification.

Using k=1, k-fold cross-validation corresponds to the leave-one-out cross-validation approach. In this method, the model is trained on n−1 samples (where n is the total number of samples) and tested on the single remaining sample. This process is repeated for all n samples, and the performance metrics are averaged, when the model is prone to overfitting, using k=1 in cross-validation may lead to higher variance in results compared to k .Overfitting Impact may cause higher accuracy on training Data: since the training set is very similar to the full dataset (only one sample is excluded), an overfitted model will likely perform very well on the training data. This might cause the model to perform less consistently across different test samples. Also, more sensitivity to outliers: with k=1, the model's test set contains only one sample. If this sample is an outlier or unusual, the performance will be heavily impacted for that iteration, leading to potentially skewed results when averaged. Predicted changes to metrics and accuracy are likely to remain high due to the large training set for each iteration [46], [47], [48]. Precision may decrease if the model struggles to generalize for rare positive cases when overfitting. Recall might still be high as overfitted models often aim to predict most cases in the dominant class correctly. Confusion matrix, the model might misclassify outliers more often, increasing false positives or false negatives. As shown in table 3.

Table 3. Performance metrics when using K-ross validation with k=1, k=10.

K	Accuracy	Recall	Precision
1	99.05%	98.84%	98.34%
10	99.10%	99.71%	99.42%

However, the information in table 3 demonstrates the efficiency when using K-ross validation with k=10. It can be perceived that the proposed model attained impressive results when the grid and ANOVA techniques were applied.

 ISSN 2723-6471

For illustration, the proposed model accomplished an exactness of 99.1, a recall of 99.4, and a precision of 99.4. the suggested model. These results endorse that the proposed model is functioning and accurate in binary classification in a validation dataset. In addition, we utilized supplementary metrics to assess the effectiveness of the suggested model. Specifically, as shown in table 4, we employed the confusion matrix based on two experiments based on K cross validation k=1, k=10.

Table 4. The evaluation of the presented model was conducted using the Confusion Matrix

K	Actual	Predicted: 0	Predicted: 1
1	0	TN = 32,548	FP = 312
	1	FN = 216	TP = 18,459
10	0	TN = 18,487	FP = 188
	1	FN = 271	TP = 32,589

The confusion matrix reveals the model's efficiency and highlights instances where the model made incorrect predictions. Specifically, it indicates the number of true positives, where the model correctly predicted the class and the sample belongs to that class, as well as true negatives, where the model appropriately predicted the class as well. As shown in table 4, the number of true positives predicted is 32,589, and the number of true negatives is 18,487. On the other hand, the confusion matrix also identifies false predictions, where the model incorrectly classifies samples that they do not belong to. These misclassifications are reflected in the false positive and false negative values. The false positives are 271, and the false negatives are 188. Table 5 represents the comparison of the accuracy of the proposed integrated and optimized model with other available ML and DL models from the collected works.

Table 5. A comparison of the accuracy of the proposed integrated and optimized model with other available ML and DL models from the collected works.

The IDEs	The core ML and DL algorithm in the model	Accuracy
[7]	IELM-APCA	70.51%.
[8]	GAwith LR	81.42%
[9]	Least Square SVM	78.86%
[10]	tRF classifier	75.66%.
[12]	RF algorithm	75.56%
[14]	CNN-Bi-LSTM	77.16%.
	The Proposed model	**99.10%**

Similarly, figure 6 disclosures of the accuracy of the obtainable proposed model per additional available ML and DL models for the binary classification task.

 ISSN 2723-6471

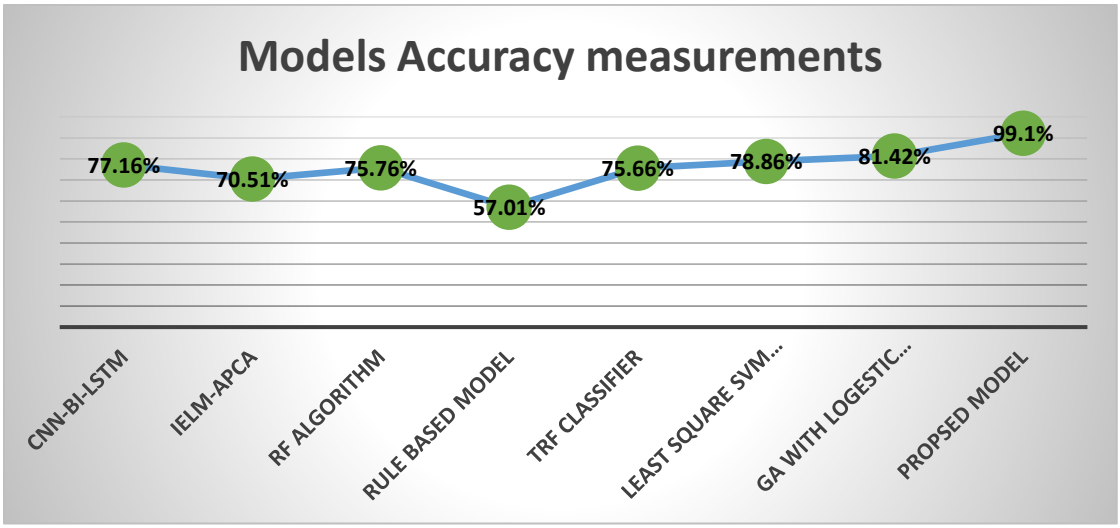

Figure 6. illustrates the accuracy comparison between the proposed model and supplementary models for binary classification Discussion.

The KNN model introduced in this study, integrating Grid Search and ANOVA algorithms, demonstrated an impressive accuracy of 99.1% in binary classification tasks using the UNSW-NB15 dataset. This performance surpassed numerous established approaches compared to the identical dataset, including the CNN-Bi-LSTM model (77.16%) [30], GA with LR model 81.42%[8], IELM-APCA (70.51%) [24], the Random Forest (RF) algorithm (75.56%) [28], LS-SVM (78.86%) [26], among others.

5. Conclusion

In summary, the proposed NIDS framework integrates ANOVA-based feature selection with grid search optimization and the KNN algorithm to tackle the complexities of network intrusion detection. Utilizing the UNSW-NB15 dataset, the model reaches an exceptional binary classification accuracy of 99.1% through meticulous hyperparameter tuning. A thorough experimental analysis, along with extensive benchmarking against advanced machine learning and deep learning techniques, clearly showcases the reliability and exceptional performance of the proposed model. The results highlight its ability to consistently outperform existing approaches, demonstrating its potential to significantly enhance the effectiveness of network security systems. This evaluation underscores the model's robustness, indicating that it can effectively address the complexities of modern network threats. Furthermore, the model's superior capabilities make it a valuable tool for improving the detection and prevention of intrusions, ensuring its practical applicability in real-world security environments. The proposed model can be integrated with existing intrusion detection systems (IDS) as a modular component. For instance, it can function as a pre-processor to filter anomalous traffic or as an additional layer of defense for classifying potentially malicious activities. However, our study focused on the K-Nearest Neighbors (KNN) algorithm, we acknowledge that other classification algorithms such as Deep Learning models can be employed. Further, the proposed model has been trained and tested on historical data the same as other current models, but it would involve integrating the model with real-time network monitoring tools, which would allow for continuous detection and rapid response to emerging threats. Also, we plan to test it on a variety of different datasets beyond the NB15 dataset. Evaluating the model on other well-known datasets, such as KDD Cup 99, CICIDS,

6. Declarations

6.1. Author Contributions

Conceptualization: M.A.A., M.A.A., R.S., T.A., R.A., and F.A.; Methodology: M.A.A.; Software: M.A.A.; Validation: M.A.A., M.A.A., and F.A.; Formal Analysis: M.A.A., M.A.A., and F.A.; Investigation: M.A.A.; Resources: M.A.A.; Data Curation: M.A.A.; Writing Original Draft Preparation: M.A.A., M.A.A., and F.A.; Writing Review and Editing: M.A.A., M.A.A., and F.A.; Visualization: M.A.A. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia (Grant No. KFU250398).

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] D. Elreedy; A. F. Atiya, “A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance,” *Information Sciences*, vol. 505, no. 3, pp. 32-64, 2019

[2] B. J. Erickson, F. Kitamura, “Magician’s Corner: 9. Performance Metrics for Machine Learning Models,” *RSNA*, vol. 3, no. 3, pp. 95-115, 2021.

[3] J. Hair and A. Alamer, "Partial Least Squares Structural Equation Modeling (PLS-SEM) in second language and education research: Guidelines using an applied example," *Res. Methods Appl. Linguistics*, vol. 1, no. 3, pp. 1-21, 2022.

[4] M. D. Firoozjaei, A. H. Lashkari, and A. A. Ghorbani, "Memory forensics tools: a comparative analysis," *J. Cyber Security Technol.*, vol. 6, no. 3, pp. 149-173, 2022.

[5] O. Sagi and L. Rokach, "Approximating XGBoost with an interpretable decision tree," *Inf. Sci.*, vol. 572, no. 10, pp. 522-542, 2021

[6] A. Nazir and R. A. Khan, "A novel combinatorial optimization-based feature selection method for network intrusion detection," *Comput. Secur.*, vol. 102, no. 17, pp. 1-27, 2021

[7] H. Zouhri, A. Idri, and A. Ratnani, "Evaluating the impact of filter-based feature selection in intrusion detection systems," *Int. J. Inf. Security*, vol. 23, no. 2, pp. 759-785, 2024

[8] J. A. Shehadeh, H. ALTaweel, and A. Qusef, "Analysis of data mining techniques on KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets for intrusion detection," *in 2023 24th Int. Arab Conf. Inf. Technol. (ACIT)*, IEEE, vol. 27, no. 5, pp. 1-6, 2023.

[9] V. Kumar, D. Sinha, A. K. Das, S. C. Pandey, and R. T. Goswami, “An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset,” *Cluster Computing*, vol.23, no. 23, pp. 1397-1418, 2020.

[10] Y. K. Saheed, A. I. Abiodun, S. Misra, M. K. Holone, and R. Colomo-Palacios, "A machine learning-based intrusion detection for detecting internet of things network attacks," *Alexandria Eng. J.*, vol. 61, no. 12, pp. 9395-9409, 2022.

[11] J. Gao, S. Chai, and B. Zhang, "Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis," *Energies*, vol. 12, no. 7, pp. 1223-1241, 2019

[12] K. Jang, W. Wang, “Network intrusion detection combined hybrid sampling with deep hierachical network,” *IEEE access*, vol. 8, no. 2, pp. 32464-32476, 2020.

[13] D. O. Howes, G. E. Chapman, “Understanding variability: the role of meta-analysis of variance,” *Cambridge University Press*, vol. 54, no. 12, pp. 3233-3236. 2024.

[14] M. D. H, Alamgir, A. Fahad; H. M. Alimul, “Towards superior android ransomware detection: An ensemble machine learning perspective,” *Cyber Security and Applications*, vol. 3, no. 1, pp. 1-16, 2025.

[15] A. Roy and K. J. Singh, "Multi-classification of UNSW-NB15 dataset for network anomaly detection system," *in Proc. Int. Conf. Commun. Comput. Technol.: ICCCT-2019, Springer Singapore*, vol. 2021, no. 1, pp. 429-451, 2021

[16] S. Wakhid, R. Sarno, S. I. Sabilla, "The effect of gas concentration on detection and classification of beef and pork mixtures using E-nose," *Comput. Electron. Agric*, vol. 195, no. 106838, pp. 195-207, 106838., 2022.

[17] A. A. Almuqren, "Cybersecurity threats, countermeasures and mitigation techniques on the IoT: Future research directions," *Journal of Cyber Security and Risk Auditing*, vol. 1, no. 1, pp. 1–11, Jan. 2025

[18] J. Paul and M. Barari, "Meta-analysis and traditional systematic literature reviews—What, why, when, where, and how?," *Psychol. Mark.*, vol. 39, no. 6, pp. 1099-1115, 2022

[19] J. G. Eisenhauer, "Meta-analysis and mega-analysis: A simple introduction," *Teach. Stat.*, vol. 43, no. 1, pp. 21-27, 2021.

[20] R. S. Mousa and R. Shehab, "Applying risk analysis for determining threats and countermeasures in workstation domain," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 1, pp. 12–21, Jan. 2025

[21] Otoom, "Risk auditing for Digital Twins in cyber physical systems: A systematic review," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 1, pp. 22–35, Jan. 2025

[22] H. Zhanhui, L. Yanping, Z. Siqing, "SAGB: self-attention with gate and BiGRU network for intrusion detection," *Complex and Intelligent Systems*, vol. 10, no. 6, pp. 8467-8479, 2024.

[23] R. Almanasir, D. Al-solomon, S. Indrawes, M. A. Almaiah, U. Islam, and M. Alshar'e,"Classification of threats and countermeasures of cloud computing," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 2, pp. 27–42, Apr. 2025

[24] M. R. Alboalebrah and S. Al-augby, "Unveiling the Causes of Fatal Road Accidents in Iraq: An Association Rule Mining Approach Using the Apriori Algorithm," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 2, pp. 1–11, Apr. 2025.

[25] E. Alotaibi, R. Bin Sulaiman, and M. Almaiah, "Assessment of cybersecurity threats and defense mechanisms in wireless sensor networks," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 1, pp. 47–59, Feb. 2025

[26] A. M. Amine and Y. I. Khamlichi, "Optimization of Intrusion Detection with Deep Learning: A Study Based on the KDD Cup 99 Database.," *International Journal of Safety & Security Engineering*, vol. 14, no. 4, pp. 1029-1038, 2024.

[27] A. Alshuaibi, M. Almaayah, and A. Ali, "Machine Learning for Cybersecurity Issues : A systematic Review," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 1, pp. 36–46, Feb. 2025

[28] M. A. Al-Shareeda, A. Mohammed Ali, M. A. Hammoud, Z. H. M. Kazem, and M. A. Hussein, "Secure IoT-Based Real-Time Water Level Monitoring System Using ESP32 for Critical Infrastructure," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 2, pp. 44–52, Apr. 2025

[29] O. Aljumaiah, W. Jiang, S. R. Addula, and M. A. Almaiah, "Analyzing Cybersecurity Risks and Threats in IT Infrastructure based on NIST Framework," *Journal of Cyber Security and Risk Auditing*, vol. 2025, no. 2, pp. 12–26, Apr. 2025

[30] Y. Yin, J. Jang-Jaccard, W. Xu, A. Singh, J. Zhu, F. Sabrina, and J. Kwak, "IGRF-RFE: A Hybrid Feature Selection Method for MLP-based Network Intrusion Detection on UNSW-NB15 Dataset," *Journal of Big Data*, vol. 10, no. 1, pp. 15-27, 2022

[31] N. L. da Costa, M. D. de Lima, and R. Barbosa, "Evaluation of feature selection methods based on artificial neural network weights," *Expert Systems with Applications*, vol. 168, no. 1, pp. 114-312, 2021

[32] E. Alotaibi, R. Bin Sulaiman, and M. Almaiah, "Assessment of cybersecurity threats and defense mechanisms in wireless sensor networks," *J. Cyber Secur. Risk Audit.*, vol. 2025, no. 1, pp. 47–59, 2025, doi: 10.63180/jcsra.thestap.2025.1.5.

[33] Ş., Halil, "Estimation of biogas potential from poultry manure to 2040 in Türkiye using time series and the artificial neural network (ANN)," *Renewable and Sustainable Energy Reviews*, vol. 207, no.1, pp. 114-967, 2025.

[34] S. Mohd Sakib, M. Suhel, and M. Alam, "Ensemble deep learning techniques for time series analysis: a comprehensive review, applications, open issues, challenges, and future directions," *Cluster Computing*, vol. 28, no. 1, pp. 28-73, 2025.

[35] Z. Zoghi and G. Serpen, "Ensemble Classifier Design Tuned to Dataset Characteristics for Network Intrusion Detection," *Electrical Engineering & Computer Science*, vol. 1, no. 1, pp. 1-41, 2022.

[36] R. H. Mohamed, F. A. Mosa, and R. A. Sadek, "Efficient Intrusion Detection System for IoT Environment," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 4, pp. 572-578, 2022

[37] A. Schwarzschild, E. Borgnia, A. Gupta, F. Huang, U. Vishkin, M. Goldblum, and T. Goldstein, "Can you learn an algorithm? Generalizing from easy to hard problems with recurrent networks," *Adv. Neural Inf. Process. Syst.*, vol. 34, no. 2, pp. 6695-6706, 2021.

[38] P. Chaudhary, B. Gupta, and A. K. Singh, "Implementing attack detection system using filter-based feature selection methods for fog-enabled IoT networks," *Telecommun. Syst.*, vol. 81, no. 1, pp. 23-39, 2022.

[39] G. Sugitha, A. Solairaj, and J. Suresh, “Block chain fostered cycle-consistent generative adversarial network framework espoused intrusion detection for protecting IoT network,” *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 11, pp. 1-18, 2022.

[40] H. Liu, Q. Li, B. Yan, L. Zhang, and Y. Gu, “Bionic electronic nose based on MOS sensors array and machine learning algorithms used for wine properties detection,” *Sensors*, vol. 19, no. 1, pp. 19-45, 2018.

[41] Y. S. Taspinar and A. Feyzioğlu, “Beef Quality Classification with Reduced E-Nose Data Features According to Beef Cut Type,” *sensors*, vol. 23, no. 4, pp. 22-22, 2023.

[42] R. Kleiman and D. Page, “AUCμ: A Performance Metric for Multi-Class Machine Learning Models,” p*roceedings of the 36th International Conference on Machine Learning, PMLR 97:3439-3447*, 2019, vol. 97, no. 3, pp. 3439-3447, 2019.

[43] M. A. Alsharaiah, A. A. Abu-Shareha, and A. Hussein, “Attention-based Long Short Term Memory Model for DNA Damage Prediction in Mammalian Cells,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 9, pp. 91-99, 2022

[44] W. I. Lai, Y. Y. Chen, and J. H. Sun, "Ensemble machine learning model for accurate air pollution detection using commercial gas sensors," *Sensors*, vol. 22, no. 12, pp. 4393-4407, 2022.

[45] Z. Liu, B. Chang, and F. Cheng, "An interactive filter-wrapper multi-objective evolutionary algorithm for feature selection," *Swarm Evol. Comput*., vol. 65, no. 4, pp. 1-25, 2021

[46] A. Rabehi, H. Helal, D. Zappa, and E. Comini, "Advancements and prospects of electronic nose in various applications: A comprehensive review," *Appl.* Sci., vol. 14, no. 11, pp. 4506-4227, 2024.

[47] L. Z. H. Jansen and K. Bennin, “machine learning algorithm for personalized healthy and sustainable grocery product recommendations,” *International Journal of Information Management Data Insights*, vol. 5, no. 1, pp. 1-13, 2025

[48] Y. Li, C. Fei, C. Mao, D. Ji, J. Gong, Y. Qin, and T. Lu, "Physicochemical parameters combined flash GC e-nose and artificial neural network for quality and volatile characterization of vinegar with different brewing techniques," *Food Chem*., vol. 374, no. 10, pp. 1-23, 202.