Zhao and Zhao
Journal of Engineering and Applied Science (2024) 71:123
https://doi.org/10.1186/s44147-024-00404-y

Journal of Engineering and Applied Science

RESEARCH Open Access

Providing a hybrid approach to increase the accuracy of intrusion detection systems in computer networks

Wei Zhao[1*] and Zhitong Zhao[1]

*Correspondence:
zhaoweidf@sina.com

[1] Yingkou Branch, Liaoning Tobacco Company, LiaoNing 115000, China

Abstract

Intrusion detection is a critical obstacle in the realm of security and data mining methodologies. Consequently, researchers have extensively investigated the quest for the swiftest and most precise means of identifying intrusions. Essentially, intrusion detection systems are tasked with recognizing any unauthorized activities, misuse, or harm inflicted upon a system, be it by internal or external users. Recently, in order to design intrusion detection systems, artificial intelligence and machine learning methods have been used, each of which has its own characteristics and advantages. Accordingly, this article focuses on using machine learning to improve the accuracy of the intrusion detection process. In fact, by using machine learning, trends, and patterns can be easily identified and thus used in a network environment to detect intrusion. It can be very useful. For this purpose, we utilize Radial Basis Function (RBF) neural networks and support vector machine (SVM) algorithm to improve decision-making and intrusion detection. By employing RBF neural networks, important features of the data are extracted, which in turnPlease check if the author details and affiliations are presented correctly. Kindly amend if necessary. enhance the overall performance of the solution and the efficiency of the SVM algorithm. This is because feature reduction ultimately leads to improved effectiveness of the SVM algorithm. In methods lacking this capability, the learning algorithm is compelled to utilize features that have no specific correlation with intrusion and essentially do not contribute to identifying attacks. Such learning approaches essentially learn from noisy data, which negatively impacts the intrusion detection solution. Finally, the proposed solution was evaluated using Python programming language and KDD99 data set. The results of the evaluation indicate that the proposed solution has a higher accuracy and precision than other evaluated solutions. So, the accuracy is 97, and the precision is over 99%.

Keywords: Intrusion detection system, Machine learning, Neural networks, KDD dataset

Introduction

With the increasing prevalence of computer networks, the security of these networks has become a top priority. Intrusion detection systems are application programs in network security infrastructures. Various methods are utilized in intrusion detection systems,

with data mining being one of the primary approaches. The implementation of intrusion detection systems is among the solutions employed to enhance security in computer networks [1]. Modern architectures employed for intrusion detection have posed challenges for system designers in selecting architectures that can provide greater reliability in detecting intrusions. In the realm of security concerns and data mining techniques, intrusion detection holds significant importance. Intrusion detection systems (IDS) can be categorized into two primary types: SIDS (signature-based Intrusion Detection System) and AIDS)anomaly-based intrusion detection system) [2].

Signature-based intrusion detection systems

SIDS systems, also known as knowledge-based detection, rely on pattern-matching techniques to identify known attacks [3]. SIDS utilizes matching methods to detect instances of intrusion that have been previously identified. When the signature of an intrusion matches with a stored signature in the database, an alarm is triggered. In SIDS, the logs of the host system are examined to identify sequences of commands or actions that have been previously recognized as malicious detection [4, 5].

Figure 1 provides a conceptual illustration of the functioning of SIDS. The main idea involves establishing a database of intrusion signatures and comparing current activities with the existing signatures. If a match is found, an alarm is raised. For example, a rule formulated as "if: antecedent - then: consequent" can be used to classify an event as an attack if the source IP address matches the destination IP address.

SIDS are renowned for their high accuracy in detecting known attacks [6]. However, they encounter challenges in detecting 0-day attacks since there are no matching signatures in the database until the signatures for new attacks are identified and stored. SIDS finds wide application in various tools like Snort and NetSTAT [7]. Several methods have been employed to create signatures for SIDS, including state machines [8, 9], formal language string patterns, or semantic conditions. The increasing prevalence of 0-day attacks [10] has diminished the effectiveness of SIDS, as there are no pre-existing signatures for such attacks. The use of AIDS, which concentrate on profiling acceptable behavior rather than identifying anomalies, as detailed in the next part, is one potential solution to this issue.

Anomaly-based intrusion detection system

AIDS systems have attracted significant attention from researchers as they address the limitations of SIDS. Using machine learning, statistical-based, or knowledge-based

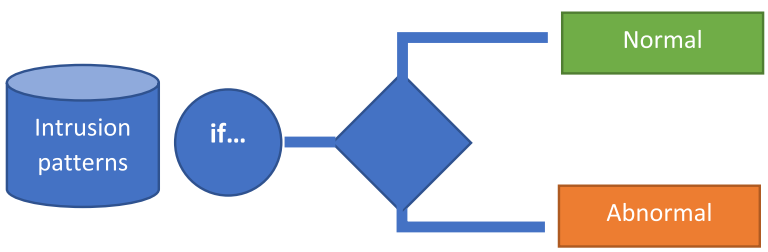

Fig. 1 IDS Concepts

techniques, a computer system's model of typical behavior in the of AIDS is created. AIDS can be categorized into various types according to the training method, such as statistical-based, knowledge-based, and machine learning-based approaches [11]. The main advantage of AIDS is its ability to detect 0-day attacks, as it does not rely on a signature database [12]. AIDS triggers an alert when the observed behavior deviates from the expected norm.

Table 1 illustrates the distinctions between anomaly-based detection and signature-based detection. SIDS can only detect known intrusions, whereas AIDS can identify 0-day attacks. However, AIDS may lead to a higher false positive rate since anomalies sometimes represent new normal activities rather than genuine intrusions.

Also a comparation between SIDS and AIDS are shown in Table 2.

This research proposes a combined intrusion detection solution based on neural networks as the base function and the support vector machine algorithm. This approach aims to provide an advanced solution for intrusion detection. The details of this method are explained in the third section. Subsequently, in the second section, the research background is examined. Then, the proposed solution is presented in the third section,

Table 1 Differences between anomaly-based detection and signature-based detection

Method	Benefits	Cons
SIDS	• SiDS effectively identifies intrusions with few false alarms (false positives). They can identify known attacks by matching patterns to a signature database. • SIDS offers quick intrusion detection. An alert signal activates when an intrusion signature matches a database signature, allowing fast response and mitigation. • SIDS excels at recognizing known attacks. They instantly identify and categorize threats based on matching patterns using a database of known signatures. • Simple design: SIDS are often simple in design. SIDS are easier to develop and administer than more sophisticated detection systems since they match observed activity to pre-defined signatures.	• Frequent signature updates: SIDS requires regular updates with new signatures to keep up with the evolving threat landscape. • Limited variant detection: SIDS is designed to detect attacks based on known signatures. • Inability to detect 0-day attacks: SIDS relies on pre-existing signatures, meaning they are ineffective in detecting 0-day attacks. • Unsuitability for multi-step attacks: SIDS are typically designed to identify individual signatures or patterns in network traffic or system logs. • Limited insight into attacks: SIDS focuses primarily on matching signatures, which can provide limited insight into the nature and context of attacks.
AIDS	• Detection of new attacks: One of the significant advantages of AIDS is its ability to detect new and previously unknown attacks. • Creation of intrusion signatures: AIDS can also help in the creation of intrusion signatures.	• Inability to handle encrypted packets: AIDS faces challenges in analyzing and detecting attacks in encrypted packets. • High false positive alarms: AIDS can generate a higher number of false positive alarms compared to signature-based intrusion detection systems (SIDS). • Difficulty in building dynamic typical profiles: Creating an accurate typical profile for a highly dynamic computer system can be challenging. • Unclassified Alerts: AIDS may generate alerts that are not clearly classified or classified, making it more challenging for security analysts to prioritize and respond to them effectively.

Table 2 Comparation between SIDS and AIDS

Ability	SIDS	AIDS
False alarm	Low	High
Detection rates	High	Low
Detect unknown attacks	Low	High
Can create attacks signature	No	Yes
Detect 0-day attack	No	Yes
Detect known attacks	No	Yes
Attack prognosis	No	Yes
Using train data	No	Yes

and finally, in the fourth section, the results obtained from evaluating the proposed solution are discussed.

Related works

Indeed, due to the importance of intrusion detection, extensive research has been conducted in this field. Researchers have explored various techniques, algorithms, and models to enhance the accuracy and effectiveness of intrusion detection systems. The aim is to develop robust and reliable methods that can identify and prevent unauthorized access, malicious activities, and network intrusions. Researchers have extensively explored machine learning techniques, including neural networks, support vector machines, decision trees, and ensemble methods, for intrusion detection. These approaches involve training models on labeled datasets to learn the patterns of normal and anomalous behavior, enabling them to detect deviations and anomalies in network traffic. In [13], a lightweight intrusion detection system is introduced, which utilizes the multilayer perceptron model. The proposed model consists of three phases. Initially, the input data undergoes preprocessing to transform it into a format suitable for machine learning and extract the required features. This operation is performed in three steps and then sent to the feature selection section. In this section, the features needed for accurate intrusion detection are selected. After passing through a filter to remove redundant information, they enter the second phase, where modeling is performed using a multilayer perceptron network. Finally, this solution was evaluated using the ADFA-LD and ADFA-WD datasets, achieving an accuracy ranging from 74 to 94%.

In [14], a solution based on deep transfer learning for feature extraction of Internet of Things (IoT) data and intrusion detection in smart cities is presented. This solution integrates deep learning models with intrusion detection technology, incorporating transfer learning techniques applicable to deep neural networks. The concept of transfer learning is applied to the integrated learning algorithm within the network that allows for addressing the problem of learning data and achieving higher accuracy in detecting non-normal data. Evaluation results demonstrate a detection accuracy of over 98%. In [15], a deep learning-based solution for anomaly detection in IoT-based industries is presented. This solution is based on data dependency in the network, enabling it to learn according to unsupervised learning models and be used on low-power devices. Unlike other methods that operate linearly, this method can also be executed in nonlinear dimensions,

providing a more powerful intrusion detection capability in the network. The solution was evaluated using the SWaT dataset, resulting in a considerable increase in the average accuracy of anomaly detection in the network, with an accuracy rate above 98%. In [16], a Bayesian network-based intrusion detection framework using packet wrapping is proposed. This article presents a two-phase framework for IDS in networks. In the first phase, a packet-wrapping method according to a genetic algorithm is used for the feature chosen. The reason for using the packet wrapping method in this research is its higher accuracy compared to other feature selection methods, such as the filter method. After constructing the model, the accuracy of the constructed model on the selected features is evaluated using the test dataset.

In the proposed method described in [17], a single-layer neural network is employed to generate fuzzy membership functions and classify unlabeled data samples. The resulting categories are then merged with the initial dataset and retrained using a classification approach. The research applies this method to the NSL-KDD dataset, and the outcomes reveal the significant influence of unlabeled data belonging to the first and third fuzzy groups on the accuracy of the classifier. In the method discussed in [18], the research investigates the impact of principal component analysis (PCA) on intrusion detection systems. It explores the determination of the optimal number of principal components required for intrusion detection and considers the effect of noisy data on PCA. The original datasets with the size of $d \times n$ are mapped to a structure with k principal components, resulting in a transformed dataset of size $k \times n$, where n represents the number of samples and d denotes the number of original dimensions. The range of variation for k is set between 2 and 20. The experimental results show classification accuracy for ten principal components at approximately 7.99% and 8.98%, which is nearly equivalent to the accuracy achieved using 41 features from 31,279 samples for the KDD dataset and 28 features from 33,746 samples for the ISCX dataset. In [19], a rule-based intrusion detection model utilizing genetic programming is introduced. The research presents a fuzzy association rule mining approach according to Genetic Network Programming (GNP) for detecting intrusions in computer networks. GNP is an evolutionary optimization method that utilizes directed graphs or trees instead of strings in genetic programming, resulting in improved solution representation power with less programming effort. The implementation results of this research on the KDD99Cup and DARPA98 datasets are examined, demonstrating its ability to achieve a reasonable level of intrusion detection accuracy.

In [20], a method called FC-ANN is proposed for intrusion detection based on ANN methods and fuzzy clustering. This method aims to achieve higher accuracy rates in intrusion detection systems. In this approach, different subsets of training data are created using fuzzy clustering in the first step. Then, based on these created subsets, various artificial neural networks are trained to create multiple models. Finally, a fuzzy aggregation method is used to summarize the results obtained from these different models. The evaluation of this research is conducted on the KDD99Cup dataset. In [21], a hidden naive Bayes (HNB) IDS is introduced for the detection of intrusions in computer networks. The researchers highlight that the HNB method is well-suited for intrusion detection problems characterized by high dimensions and interdependent features. HNB is a data mining approach that adjusts the assumptions of the naive Bayes method.

The suggested method is implemented and assessed using the KDD dataset, showcasing the substantial advantage of the HNB method over the naive Bayes method in terms of enhancing accuracy and reducing error rates. The research presented in [22] introduces an intrusion detection model that combines chi-square feature selection with multi-class support vector machines. Many existing IDS rely on a single classification algorithm to categorize network traffic as normal or abnormal. However, according to the vast amount of data, these models often face challenges in achieving high detection rates while simultaneously reducing false alarm rates. To address this, the proposed model incorporates square-chi feature selection and utilizes multi-class SVM to improve the performance of intrusion detection by enhancing detection rates and minimizing false alarms. However, in this approach, by reducing the dimensionality of the data, an optimal set of features is obtained without losing information. Then, using the multi-class modeling method, different types of network attacks are classified. In [23], a solution for data classification in order to detect intrusions in computer networks is proposed by combining support vector machine (SVM) methods. The goal of this method is to classify normal and abnormal data with high accuracy and reduce the error rate. In this article, the combination of SVM with a self-organizing ant colony clustering network is presented. The proposed method is evaluated using the KDD99Cup dataset to assess its effectiveness in intrusion detection. The results indicate the superiority of the combined approach over each individual method of SVM and self-organizing ant colony clustering network. Table 3 presents a comparison of the existing methods in the research background.

Proposed method

In this section, details of the proposed hybrid method are presented. To implement the proposed solution, the suggested model in Fig. 2. is utilized in the networks. The following provides an explanation of the details of each section. In the proposed method, before applying the data to the suggested model, a preprocessing step is performed to normalize the data. This section includes the following stages:

Table 3 Comparison of the existing methods in the related works

Dataset	Learning method	Method	IDS type	Environment	Ref.
ADFA-LD	Supervised	Multilayer perceptron neural network	Signature-based	Fog and IoT	[13]
KDD99Cup	Semi-supervisory	Deep neural network	Signature-based	Fog and IoT	[14]
SWaT	Unsupervised	Cryptography based on neural network	Anomaly-based	Fog and IoT	[15]
–	Supervised	Bayesian network and genetics	Anomaly-based	Network	[16]
NSL-KDD	Semi-supervisory	Fuzzy	Signature-based	Network	[17]
KDDCUP		PCA	Signature-based	Fog and IoT	[18]
KDD99Cup	Semi-supervisory	Fuzzy	Anomaly-based	Fog and IoT	[19]
KDD99Cup	Supervised	Neural network and fuzzy clustering	Signature-based	Fog and IoT	[20]
KDD	Unsupervised	Naive Bayesian multiple classification	Signature-based	Fog and IoT	[21]
NSL-KDD	Supervised	Support vector machine	Anomaly-based	Fog and IoT	[22]
KDD99Cup	Supervised	Support vector machine and ant colony	Anomaly based	Network	[23]

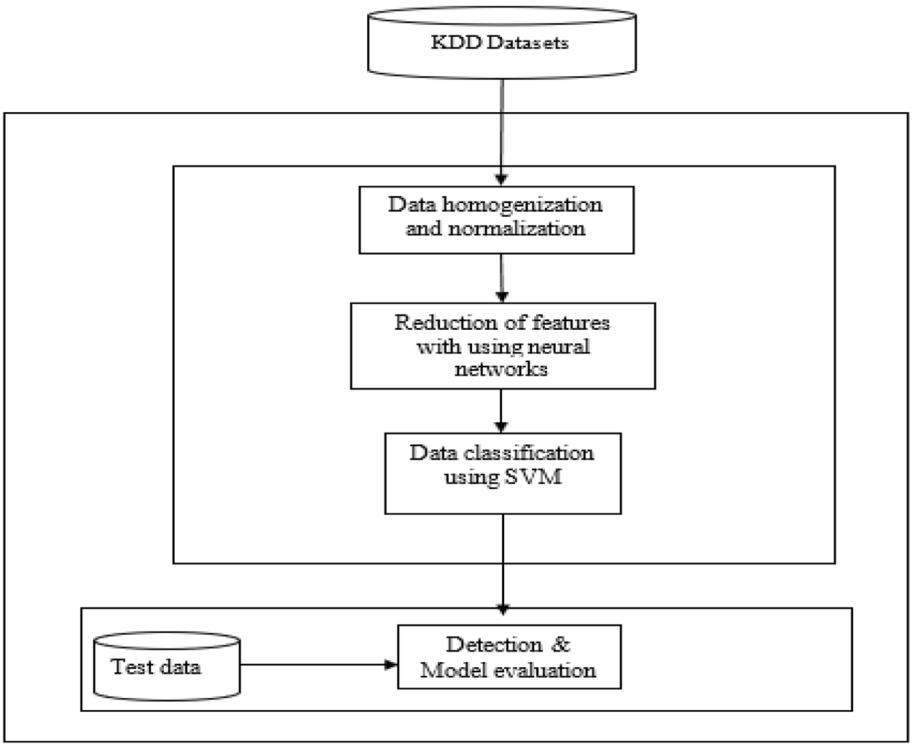

Fig. 2 Proposed method

- Data homogenization (training and testing data): In this stage, homogenization is applied to the data at the data level by replacing alphabetical characters in the dataset with numerical values of a specific type.
- Data standardization (training and testing data): In this stage, the data imbalance is addressed. Since the KDDcup99 dataset is used, some features have large numerical values that can dominate other features. In this section, this type of anomality is eliminated.

In the next module, the operations related to the neural network are performed. Specifically, by utilizing Radial Basis Function (RBF) neural networks, important features of the data are extracted to enhance the overall performance of the solution and the efficiency of the support vector machine (SVM) algorithm. Feature reduction can ultimately improve the efficiency of the SVM algorithm. In methods that lack this capability, the learning algorithm is forced to use features that have no specific relevance to intrusion and, in other words, do not play a role in determining attacks. This type of learning essentially involves learning with noisy data and has a negative impact on the intrusion detection solution.

In the proposed solution, the support vector machine (SVM) classifier algorithm is used for learning, but instead of using all features, it only utilizes the important features provided by the neural network component. This leads to improved performance in the SVM algorithm. On an end of a statement and a start of a new statement, processing, deployment, and utilization of features that do not play a significant role in classification decrease

intrusion detection and learning speed. Neural networks perform a mapping from raw data space to conceptual space by reducing the data dimensionality and selecting important features. Some of the main reasons for choosing RBF neural networks include:

- It provides a faster solution compared to other methods, and the computational complexity of previous algorithms is reduced.
- Unlike multilayer perceptron MLP neural networks, where synaptic weights of all layers need to be calculated, in RBF networks, the input layer is connected to the hidden layers without synaptic weights between layers.
- In this algorithm, the neurons in the hidden layers act as a nonlinear kernel (Gaussian RBF) and are responsible for mapping the data from a nonlinear space to a linear space.

Based on these reasons, the following are the details of each component in the proposed solution.

Feature reduction

After preprocessing the data, in this stage, Radial Basis Function (RBF) neural networks are used to perform the learning process on the obtained dataset and reduce the features. RBF neural networks are one of the most powerful types of neural networks used in function approximation problems. They are more capable than perceptron neural networks and have functions in the areas of learning, identification, similarity, and control of dynamic nonlinear systems. These functions are used in the estimation and prediction of time series. In an artificial neural network, these functions can be employed as activation functions for neurons. However, unlike perceptron networks that typically have multiple layers, the Radial Basis Function (RBF) neural network consists of three fixed layers [24]. The input layer is where the input signals are injected into the network, the middle layer contains RBF functions, and the output layer is a linear combination of all the outputs from the middle layer. In most cases, the Gaussian function is used in the middle layer, where these functions are identified by two parameters: the Gaussian center and the variance or spread of the Gaussian. Figure 3 illustrates an example of an RBF neural network.

In these networks, the hidden layer performs a vital function by establishing a nonlinear mapping between the input space and typically a higher-dimensional space. Its role is to transform nonlinear patterns into linearly separable patterns. On the other hand, the output layer combines the weighted linear patterns with a linear output. This output is particularly useful when using RBF for function approximation. However, if the goal is pattern classification, the output neurons can be equipped with a hard limiter or sigmoid function to generate output values of 0 or one [25]. From the aforementioned descriptions, it is evident that the hidden layer is where the unique processing of the network occurs. The hidden layer function is described by the following eq. (1):

$$F(x) = \sum_{j=1}^{p} W_j \varnothing(\|x - uj\|) \tag{1}$$

In the above equation, *Wj* represents the weights associated with each neuron, and *uj* represents the centroids of the neurons. The primary function used in these networks is the Gaussian function, which is expressed by Eq. (2):

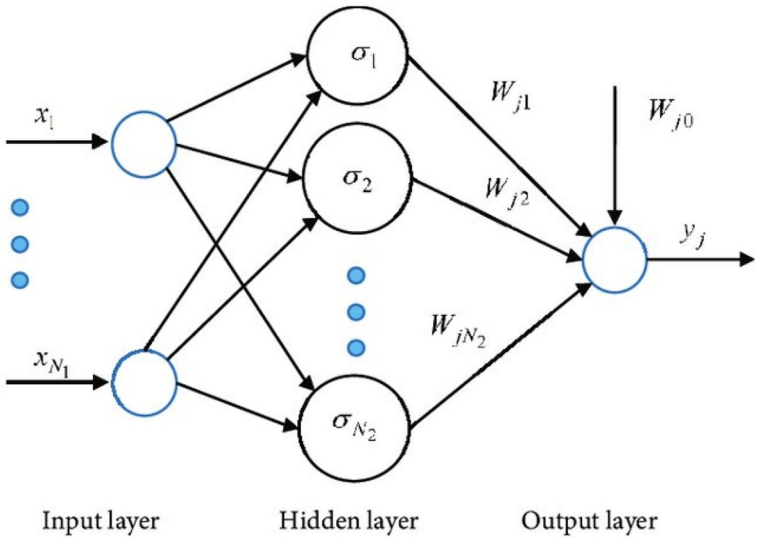

Fig. 3 An example of an RBF neural network

$$\varnothing\left(\|x - uj\|\right) = e^{\frac{(\|x-uj\|)}{\sigma}} \tag{2}$$

In the above equation, σ is the kernel width. Similar to other neural networks, these networks have two modes: learning and recall. In the learning mode, the network's adaptive parameters, including uj, σ, and the output layer weight matrix W, are adjusted in such a way that the average error between the network's outputs on a training dataset and the actual values is minimized. In the recall mode, the trained network generates output vectors by providing new input vectors to the network. Let the numbers of normal and assault samples in the training set be m+ and m−, respectively. Let the false-alarm and false-classification rates for the normal and attack samples be $\epsilon +$ and $\epsilon -$, respectively (the sum of the false-classification and detection rates = "1"). Next, the following is the definition of a weighted mean ϵ of the classification errors:

$$\epsilon = \frac{1}{m}\sum_{j=1}^{m} I\left(h[yj] \neq xj\right) = \frac{1}{m}[j : xj = +1, h[yj] = -1] \tag{3}$$

Support vector machine

The SVM algorithm is a supervised non-parametric statistical method that operates under the assumption that there is no knowledge about the distribution of the dataset. Its main feature is the ability to achieve high accuracy with fewer training samples compared to other classification methods [26]. In Fig. 4, two classes and their support vectors are depicted. It is assumed that the data consists of two classes, denoted by xi (i = 1,..., L), where xi represents a vector and the classes are labeled as $yi = \pm 1$. Since we aim to divide the data into two categories: clean and contaminated, these two classes are considered completely separate. To determine the decision boundary that separates the two classes completely, the method of optimal margin is used. In the SVM method, a linear boundary is determined in such a way that the following conditions are satisfied:

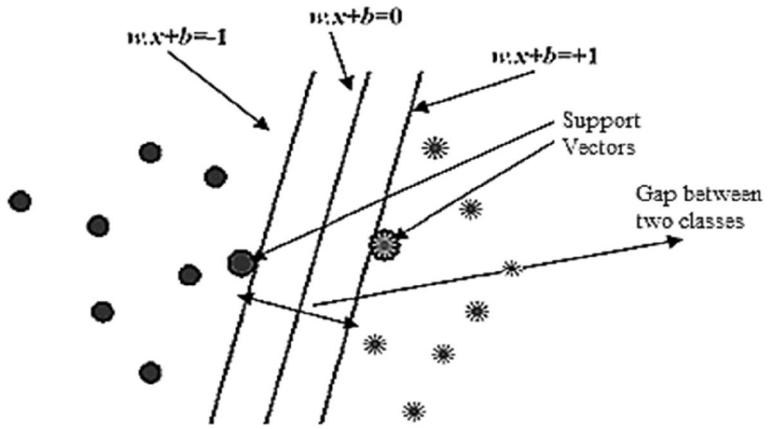

Fig. 4 Optimal linear boundary where two classes are completely separated from each other

- All samples belonging to class $+1$ are positioned on one side of the decision boundary, while all samples of class -1 are situated on the other.
- The decision boundary is calculated in a manner that maximizes the distance between the closest training samples of both classes, which is done perpendicular to the decision boundary.

A linear decision boundary can generally be written as:

$$w.x + b = 0$$

In the above equation, x denotes a point situated on the decision boundary. w represents an n-dimensional vector that is orthogonal to the decision boundary, and b indicates the distance from the origin to the decision boundary. The term $w.x$ corresponds to the dot product between the vectors w and x. Since multiplying both sides of the equation by a constant still holds equality, the following conditions are imposed to define the unique values of b and w:

$$\mathrm{If}xi\mathrm{isasupportvector} : y_i(w.x + b) = 0 \tag{4}$$

$$\mathrm{If}xi\mathrm{isnotasupportvector} : y_i(w.x + b) < 1 \tag{5}$$

The initial step in determining the optimal decision boundary involves identifying the nearest training samples from the two classes. Subsequently, the distance between these points is measured in the direction perpendicular to the boundaries that entirely separate the two classes. This distance calculation aids in finding the optimal margin, which maximizes the separation between the classes. The optimal decision boundary is the one that maximizes the margin. In fact, methods like SVM attempt to separate the data by constructing a hyperplane. SVM, as a linear classification method, finds the best hyperplane that maximally separates the data points belonging to the two classes with the largest margin.

Separating hyperplane

To form the separating hyperplanes, let us explain the process in detail using an example. A precise illustration of how Support vector machines form the separating hyperplanes is shown in Fig. 5.

Initially, a convex hull is constructed around the points of each class. In Fig. 4, convex hulls are depicted, enclosing the points of class -1 and class $+1$. Line P represents the closest distance line between the two convex hulls. The line h, which serves as the separating hyperplane, bisects line P and is perpendicular to it. The offset b corresponds to the distance from the origin to the separating hyperplane with the maximum margin. Ignoring b would result in hyperplanes solely passing through the origin [27, 28].

The perpendicular distance from the separating hyperplane to the origin is computed by dividing the absolute value of the parameter b by the length of the vector w. The fundamental concept is to select an appropriate separator that exhibits the maximum distance from the neighboring points of both classes. This solution essentially possesses the maximum margin with the points associated with the two distinct classes and can be bounded by two parallel hyperplanes that pass through at least one point from each class. These vectors are referred to as support vectors. The mathematical formulation of these two parallel hyperplanes forming the separating boundary is illustrated in Eqs. (6) and (7):

$$\text{Hyperplane1} : w.x1 + b = d1 \tag{6}$$

$$\text{Hyperplane2} : w.x2 + b = d2 \tag{7}$$

It is crucial to acknowledge that in cases where the training data is linearly separable, it is feasible to select two boundary hyperplanes in a manner that no data points reside between them. The objective is then to maximize the distance ($d1$–$d2$) between these parallel hyperplanes. Utilizing geometric principles, the distance between these two parallel hyperplanes is determined as $d/|w|$. Consequently, the goal becomes minimizing the value of $|w|$. Moreover, it is essential to ensure that the data points do

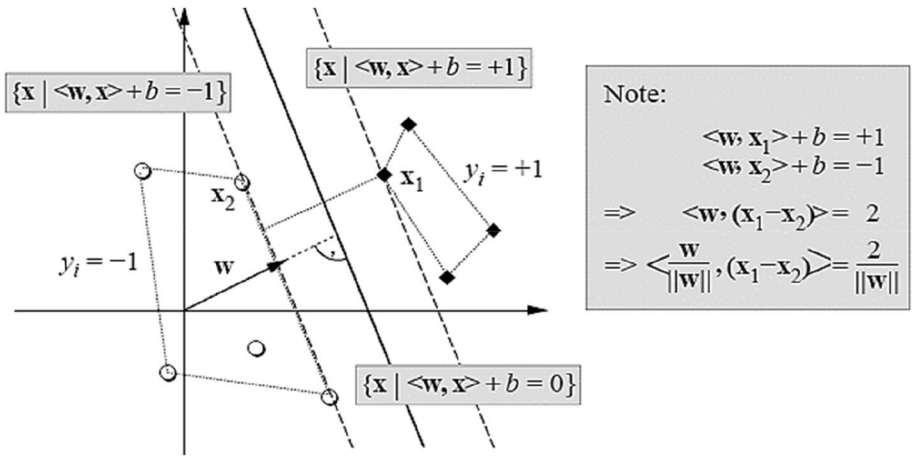

Fig. 5 Formation of separating hyperplanes between classes

not fall within the margin region. To address this concern, a mathematical constraint is incorporated into the formal definition.

For each i, the following constraints are imposed to ensure that no points lie on the boundary:

$$yi(w.x1i + b) > (d1 + d2)/2$$
$$yi(w.x2i + b) < (\mathrm{d1} + \mathrm{d2})/2$$

These constraints state that the product of the class label yi and the value of $(w.xi + b)$ should be greater than or equal to 1 for each training example xi. These constraints ensure that all data points are correctly classified and lie outside the margin region.

By solving the optimization problem with these constraints, the SVM algorithm determines the optimal values of w and b, leading to the selection of the separating hyperplanes with the maximum margin.

The constraints can be expressed in the Eq.(8):

$$c_i(w.x_i - b) \geq 1, 1 \leq i \leq n \leq n \tag{8}$$

Thus, the optimization problem can be defined as minimizing w/d while satisfying the following constraints:

$$c_i(w.x_i - b)1 \leq i \leq n$$

The objective is to find the values of w and b that minimize w while ensuring that all training examples are correctly classified and lie outside the margin region.

Methods

In this research, Python programming language has been used as a tool for implementing and testing the proposed model for anomaly detection. Python is a fast-executing language that is well-suited for processing large-scale data. To evaluate the solution, the KDD99 dataset has been used. The KDD99 dataset consists of 41 extracted features for each connection, which indicate the connection's status label, either normal or belonging to a specific class of attack [29, 30]. These features generally have continuous, discrete, and symbolic states with a wide range of values.

Evaluation

The confusion matrix is commonly used as a prevalent tool for evaluating and assessing the performance of a classification model in intrusion detection systems. It is a straightforward matrix that encompasses the following cases: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). This matrix, typically represented as a 2×2 dimensional matrix, illustrates the association between the actual values and the predicted values [31], as shown in Table 4.

In Table 5, the data points within the desired class are labeled as "Positive," and the data points outside the desired class are labeled as "Negative." The second row of this table indicates that the data points are within the desired class, and the third row indicates that the data points are outside the desired class. The middle column corresponds to the prediction or the result of the proposed method for classifying the desired data. The proposed method has classified the desired data point as

Table 4 Functions of classification rates

Predicted value			
Negative	**Positive**		
FN	TP	Positive	Actual Value
TN	FP	Negative	

Table 5 Parameters value

Parameter	Value
Value of C parameter	[0.002, 100]
Kernel parameter γ	[0.0002, 70]
No. of neurons	13
No. of hidden layers	1
Max of iterations	100

being inside the class. The column on the left indicates that the proposed method has detected and classified the desired data point as being outside the class.

Evaluation criteria

The proposed solution is evaluated according to the criteria of precision, F1, accuracy and recall. The formula for these relations is as follows:

Precision is a metric that quantifies the ratio of true positive predictions to the total number of positive predictions made by a classification model.

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP})$$

Accuracy is a metric that calculates the ratio of correctly identified true positives and true negatives to the total number of data points.

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

Recall calculates the ratio of true positive predictions to the total number of actual positives.

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN})$$

The F1 score provides a balanced assessment of a classification model's performance by considering both precision and recall simultaneously. The harmonic mean ensures that the F1 score gives equal weight to precision and recall, making it a useful metric for evaluating models when both precision and recall are important.

$$\text{F1score} = 2 \times (\text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall})$$

To evaluate the performance of the proposed solution, it has been compared and evaluated against the NBFS, NBTree, and C4.5 algorithms [32, 33]. Additionally, in Table 5, the values of the parameters used in the evaluation process are presented.

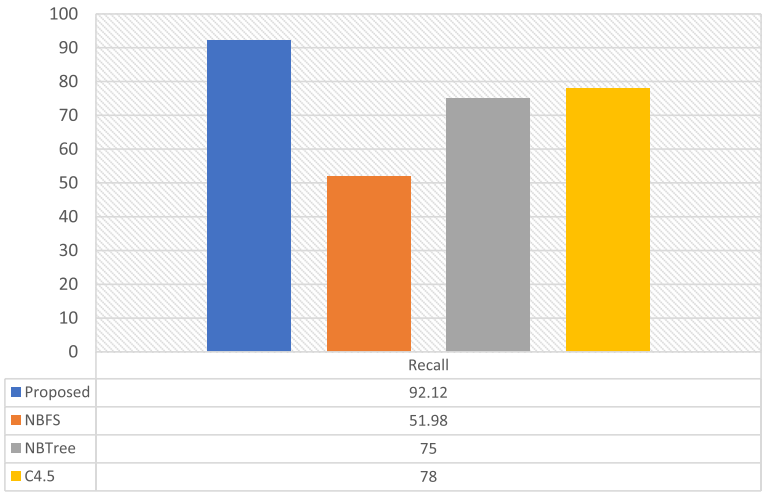

Fig. 6 Comparison of the recall parameter among the evaluated solutions

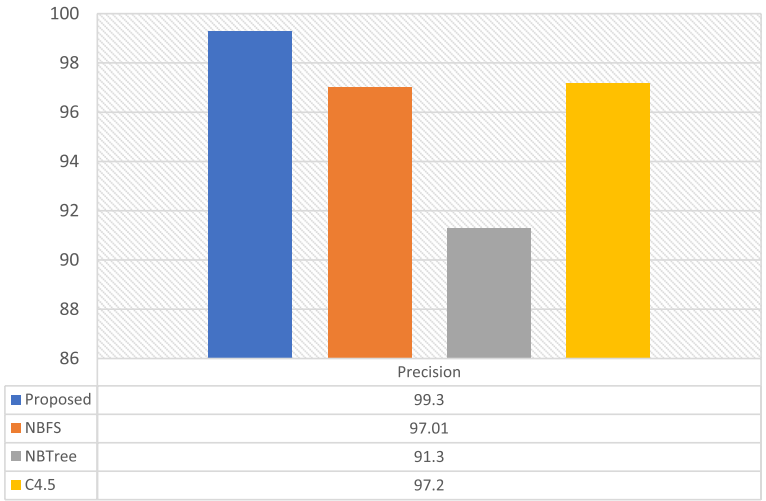

Fig. 7 Comparison of the specificity parameter among the evaluated solutions

Results and discussion

In the following section, the results of the evaluation are shown in Figs. 6, 7, 8 and 9. Firstly, in Fig. 5, the recall rates of the solutions are displayed and compared. This parameter represents the number of true positive cases correctly predicted out of the total number of positive cases. It assesses how well the proposed methods classify the data points that belong to the desired class. As observed, the proposed solution outperforms the NBFS, C4.5, and NBTree solutions. This level of effectiveness in accurately identifying intrusions is achieved through the proper utilization of the combined method, specifically by employing the RBF neural network. This approach has successfully eliminated unnecessary features for intrusion detection, resulting in improved accuracy in identifying positive cases.

Continuing with Fig. 7, the accuracy rates of the solutions are displayed. This parameter is used to assess how well the proposed method classifies the data points that do not

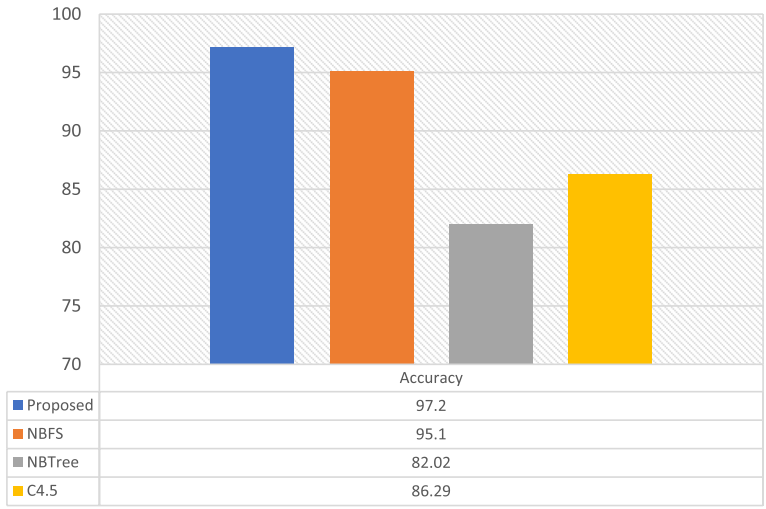

Fig. 8 Comparison of accuracy rates among evaluated solutions

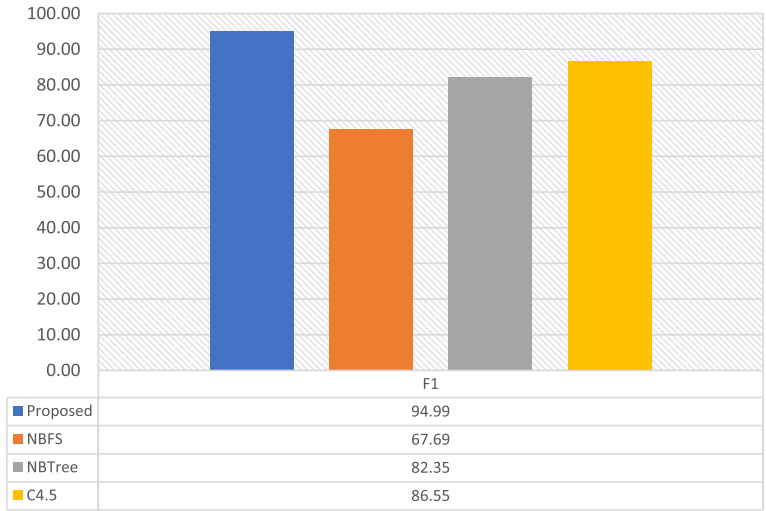

Fig. 9 Comparison of F1 measure among evaluated solutions

belong to the desired class as being within the class. As observed, the proposed solution has successfully classified the data points into the correct classes. In fact, it has achieved an accuracy rate of over 99%. The main reason for this high level of performance lies in the effective utilization of RBF neural networks, which have been able to select suitable datasets for the SVM algorithm. This enables the best learning to take place by focusing on the most important data points for intrusion detection, as indicated by the obtained results.

In Fig. 8, the accuracy rates of the evaluations are shown. As observed, the proposed solution outperforms the other three solutions. The accuracy of intrusion detection operations has reached over 97%, which is more than a 10% improvement compared to the NBTree and C4.5 solutions. This indicates a lower error rate alongside accurate detection.

In Fig. 9, the solutions are compared based on the F-score parameter. The F-measure metric is used to assess the accuracy level of detection operations using different classifiers. This metric is related to precision (P) and recall (R), and as the classifier's precision increases, the F-measure also increases. As observed, the proposed solution is more optimal compared to other evaluated solutions. By examining the results, it can be observed that feature selection has significantly improved the classifier's performance compared to methods that use all features. Overall, considering the F-measure results for all methods, the proposed solution with feature selection through the utilization of neural networks has achieved higher rates.

In addition, the results of classifying the test data are presented in Fig. 10.

Although classification using the SVM algorithm and various kernels has shown good performance, the overall evaluation results indicate that employing neural networks alongside the SVM algorithm yields better results. As observed in Fig. 5, the proposed solution, which combines the SVM algorithm and neural networks, has achieved the best results. In other words, the presented solution is capable of effectively dealing with new attacks, as its detection rate exceeds 99%. The proposed system not only selects an optimal feature set but also determines optimal parameters for the kernel function in the SVM classifier by reducing the features. Consequently, it reduces the number of features

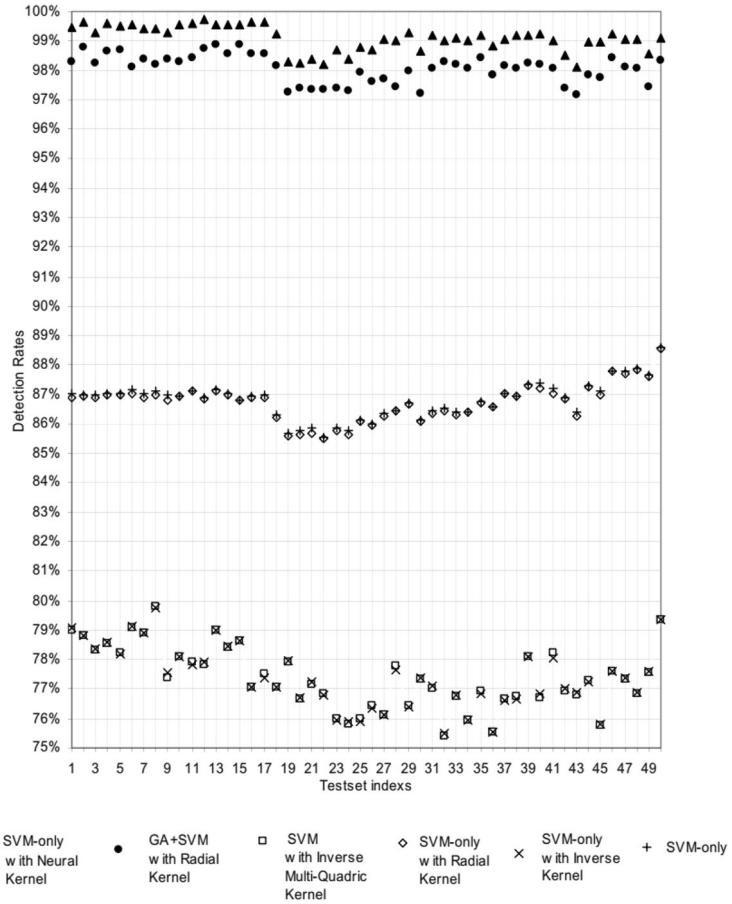

Fig. 10 The results of data classification

that the SVM needs to process and maximizes the detection rate for IDS systems in network environments. Additionally, as evident in Fig. 5, the proposed solution offers a significantly higher detection rate compared to a system that solely relies on SVM for intrusion detection.

Evaluation of execution time

In this section, the proposed solution's execution time is assessed and compared to other approaches. The evaluation aims to determine the efficiency and computational performance of the proposed solution in relation to alternative methods or techniques. By analyzing the execution time, insights can be gained into the speed and efficiency of the proposed solution, providing valuable information for decision-making and performance comparisons. The following approaches have been selected for comparison with the presented method [12]:

- Support vector machine (SVM)
- Naive Bayes algorithm
- Gradient Boosting Tree algorithm

Figure 11 presents the evaluation results. It should be noted that the dataset used is the same KDDcup99 dataset.

The evaluation results, as depicted in Fig. 10, demonstrate a notable reduction in the execution time of the proposed method compared to the other three approaches. The graph visually illustrates the significant time advantage offered by the proposed method over the alternative approaches. In comparison to the SVM-based intrusion detection solution, the detection time is faster, and it is also much shorter compared to the other two approaches. The reason for this level of efficiency lies in the utilization of the RBF neural network, which extracts important features from the data. By employing this algorithm, the overall performance of the solution and the effectiveness of the support vector machine (SVM) algorithm is enhanced. Feature reduction ultimately leads

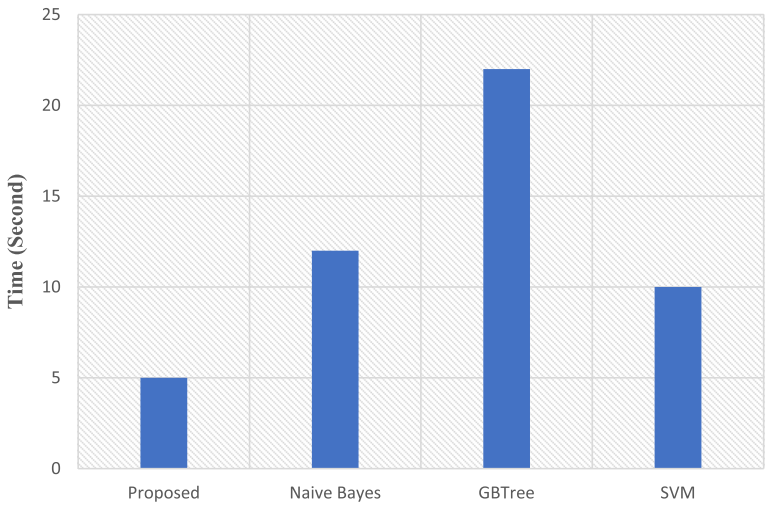

Fig. 11 Comparison of the execution time of solutions

to increased efficiency of the SVM algorithm. In methods that lack this capability, the learning algorithm is forced to use features that have little or no specific correlation with intrusions and, in essence, do not contribute to the identification of attacks. This type of learning essentially learns from noisy data, which negatively impacts the intrusion detection solution. As observed in the evaluation results, other methods, including SVM alone, do not exhibit satisfactory performance and, as a result, have relatively higher execution times.

Conclusions

By comparing these parameters, it can be concluded that through the utilization of the proposed combined solution, we have successfully optimized the key parameters for intrusion detection in a network, allowing for accurate intrusion detection using the optimized support vector machine (SVM) algorithm empowered by RBF neural networks. The outputs from previous stages are fed as inputs to the classifier algorithm, enabling the classification of detected intrusions into specific classes. Based on these classes, the intrusion detection system can make necessary decisions. If the classification at this stage is not accurate, the effectiveness of the solution will be compromised. As observed, the proposed solution has achieved an accuracy rate exceeding 97%, which is significantly higher compared to the accuracy rates of 82% and 86% achieved by the other two solutions.

Acknowledgements
I would like to take this opportunity to acknowledge that there are no individuals or organizations that require acknowledgment for their contributions to this work.

Authors' contributions
WZ performed data collection, simulation, and analysis. ZZ evaluate the first draft of the manuscript, editing, and writing.

Funding
This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Availability of data and materials
Data can be shared upon request.

Declarations

Competing interests
The authors declare no competing interests.

Received: 9 October 2023 Accepted: 28 February 2024
Published online: 28 May 2024

References

1. Lansky J et al (2021) Deep learning-based intrusion detection systems: a systematic review. IEEE Access 9:101574–101599
2. Khraisat A, Gondal I, Vamplew P, Kamruzzaman J (2019) Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity 2(1):1–22
3. Wang M, Zheng K, Yang Y, Wang X (2020) An explainable machine learning framework for intrusion detection systems. IEEE Access 8:73127–73141
4. Mulyanto M, Faisal M, Prakosa SW, Leu J-S (2020) Effectiveness of focal loss for minority classification in network intrusion detection systems. Symmetry (Basel) 13(1):4
5. Sun J, Zhang Y, Trik M (2022) PBPHS: a profile-based predictive handover strategy for 5G networks. Cybern Syst 86–99
6. Meng C, Motevalli H (2024) Link prediction in social networks using hyper-motif representation on hypergraph. Multimed Syst 30(3):123
7. Gümüşbaş D, Yıldırım T, Genovese A, Scotti F (2020) A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems. IEEE Syst J 15(2):1717–1731

8. Magán-Carrión R, Urda D, Díaz-Cano I, Dorronsoro B (2020) Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches. Appl Sci 10(5):1775
9. Trik M, Akhavan H, Bidgoli AM, Molk AMNG, Vashani H, Mozaffari SP (2023) A new adaptive selection strategy for reducing latency in networks on chip. Integration 89:9–24
10. Asghari A, Zoraghchian AA, Trik M (2014) Presentation of an algorithm configuration for network-on-chip architecture with reconfiguration ability. International Journal of Electronics Communication and Computer Engineering (IJECCE) 5(5):124–136
11. Kadir DH (2021) Statistical evaluation of main extraction parameters in twenty plant extracts for obtaining their optimum total phenolic content and its relation to antioxidant and antibacterial activities. Food Sci Nutr 9(7):3491–3499
12. Chang V et al (2022) A survey on intrusion detection systems for fog and cloud computing. Future Internet 14(3):89
13. Karabulut E, Gholizadeh F, Akhavan-Tabatabaei R (2022) The value of adaptive menu sizes in peer-to-peer platforms. Transp Res Part C Emerg Technol 145:103948
14. Li D, Deng L, Lee M, Wang H (2019) IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning. Int J Inf Manag 49:533–545
15. Trick M, Boukani B (2014) Placement algorithms and logic on logic (LOL) 3D integration. J Math Comput Sci 8(2):128–136
16. Wang G, Wu J, Trik M (2023) A novel approach to reduce video traffic based on understanding user demand and D2D communication in 5G networks. IETE J Res 1–17
17. Saleh DM, Kadir DH, Jamil DI (2023) A comparison between some penalized methods for estimating parameters: simulation study. QALAAI ZANIST J 8(1):1122–1134
18. Vasan KK, Surendiran B (2016) Dimensionality reduction using principal component analysis for network intrusion detection. Perspect Sci (Neth) 8:510–512
19. Sajadi SM, Kadir DH, Balaky SM, Perot EM (2021) An eco-friendly nanocatalyst for removal of some poisonous environmental pollutions and statistically evaluation of its performance. Surf Interfaces 23:100908
20. Kadir D. (2018) Bayesian inference of autoregressive models (Doctoral dissertation, University of Sheffield).
21. Khezri E, Bagheri-Saveh MI, Kalhor MM, Rahnama M, Roshani D, Salehi K (2022) Nursing care based on the Support-Based Spiritual Care Model increases hope among women with breast cancer in Iran. Support Care Cancer 30:423–429
22. Fakhri PS, Asghari O, Sarspy S, Marand MB, Moshaver P, Trik M (2023) A fuzzy decision-making system for video tracking with multiple objects in non-stationary conditions. Heliyon 9(11):422–446
23. Xiao L, Cao Y, Gai Y, Khezri E, Liu J, Yang M (2023) Recognizing sports activities from video frames using deformable convolution and adaptive multiscale features. J Cloud Comput 12(1):1–20
24. Wang Z, Jin Z, Yang Z, Zhao W, Trik M (2023) Increasing efficiency for routing in internet of things using binary gray wolf optimization and fuzzy logic. J King Saud Univ-Comput Inform Sci 35(9):101732
25. Ding X, Yao R, Khezri E (2023) An efficient algorithm for optimal route node sensing in smart tourism urban traffic based on priority constraints. Wireless Networks 124–131
26. Trik M, Pour Mozafari S, Bidgoli AM (2021) An adaptive routing strategy to reduce energy consumption in network on chip. J Adv Comput Res 12(3):13–26
27. MokhlesiGhanevati D, Khorami E, Boukani B, Trik M (2020) Improve replica placement in content distribution networks with hybrid technique. J Adv Comput Res 11(1):87–99
28. Khezri E, Zeinali E, Sargolzaey H (2023) SGHRP: secure greedy highway routing protocol with authentication and increased privacy in vehicular ad hoc networks. Plos One 18(4):e0282031
29. Zhang L, Hu S, Trik M, Liang S, Li D (2024) M2M communication performance for a noisy channel based on latency-aware source-based LTE network measurements. Alex Eng J 99:47–63
30. Sameera N, Shashi M (2019) Intrusion detection analytics: a comprehensive survey. Int J Adv Sci Res Manag (IJASRM) 4(6):2455–6378
31. Khezri E, Yahya RO, Hassanzadeh H, Mohaidat M, Ahmadi S, Trik M (2024) DLJSF: Data-Locality Aware Job Scheduling IoT tasks in fog-cloud computing environments. Results Eng 21:101780
32. Zhu J, Hu C, Khezri E, Ghazali MMM (2024) Edge intelligence-assisted animation design with large models: a survey. J Cloud Comp 13(1):48
33. Samiei M, Hassani A, Sarspy S, Komari IE, Trik M, Hassanpour F (2023) Classification of skin cancer stages using a AHP fuzzy technique within the context of big data healthcare. J Cancer Res Clin Oncol 149(11):8743–8757