Data and Metadata. 2025; 4:474
doi: 10.56294/dm2025474

ORIGINAL

Enhancing Multiclass Network Intrusion Detection Systems Using Continuous Wavelet Transform on Network Traffic

Mejora de los sistemas multiclase de detección de intrusos en red mediante la transformada de ondícula continua en el tráfico de red

Abdulaziz A. Alsulami[1], Badraddin Alturki[2]

[1]Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia.
[2]Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia.

Cite as: Alsulami AA, Alturki B. Enhancing Multiclass Network Intrusion Detection Systems Using Continuous Wavelet Transform on Network Traffic. Data and Metadata. 2025; 4:474. https://doi.org/10.56294/dm2025474

Submitted: 29-02-2024 **Revised:** 21-07-2024 **Accepted:** 04-11-2024 **Published:** 01-01-2025

Editor: Adrián Alejandro Vitón-Castillo

Corresponding author: Abdulaziz A. Alsulami

ABSTRACT

Network systems are susceptible to cyberattacks, which motivates attackers to exploit their vulnerabilities. Scanning network traffic to identify malicious activity is becoming a trend in the cybersecurity domain to mitigate the negative effects of intruders. Network intrusion detection systems (NIDS) are widely recognized as essential tools against cyberattacks. However, there is a need to go beyond designing traditional NIDS, which are preferred to be used with binary classification, towards designing multiclass network intrusion detection systems (MNIDS) to predict the cyberattack category. This, indeed, assists in understanding cyberattack behavior, which mitigates their effects quickly. Machine learning models, including conventional and deep learning, have been widely employed in the design of MNIDS. However, MNIDS based on machine learning can face challenges in predicting the category of cyberattack, especially with complex data that has a large number of categories. Thus, this paper proposes an enhanced MNIDS by exploiting the power of integrating continuous wavelet transform (CWT) with machine learning models to increase the accuracy of predicting cyberattacks in network traffic. This is due to the fact that CWT is considered as an effective method for feature extraction. The experimental results emphasize that using CWT with machine learning models improves the classification performance of MNIDS by up to 3,36 % in overall accuracy. Additionally, it enhances the F1-score value in up to 40 % of the total classes using the proposed model.

Keywords: Machine Learning; Deep Learning; Intrusion Detection System; Cyberattacks; Continuous Wavelet Transform.

RESUMEN

Los sistemas de red son susceptibles de sufrir ciberataques, lo que motiva a los atacantes a explotar sus vulnerabilidades. Escanear el tráfico de red para identificar actividades maliciosas se está convirtiendo en una tendencia en el ámbito de la ciberseguridad para mitigar los efectos negativos de los intrusos. Los sistemas de detección de intrusiones en la red (NIDS) están ampliamente reconocidos como herramientas esenciales contra los ciberataques. Sin embargo, es necesario ir más allá del diseño de los NIDS tradicionales, que se utilizan preferentemente con clasificación binaria, hacia el diseño de sistemas de detección de intrusiones en red multiclase (MNIDS) para predecir la categoría del ciberataque. Esto, de hecho, ayuda a comprender el comportamiento de los ciberataques, lo que mitiga sus efectos rápidamente. Los modelos de aprendizaje automático, incluido el aprendizaje convencional y profundo, se han empleado ampliamente en el diseño de MNIDS. Sin embargo, los MNIDS basados en el aprendizaje automático pueden enfrentar desafíos en la predicción de la categoría de ciberataque, especialmente con datos complejos que tienen un gran número de categorías. Por ello, este artículo propone un MNIDS mejorado que explota el poder de la integración de la transformada wavelet continua (CWT) con modelos de aprendizaje automático para aumentar la precisión de la predicción de ciberataques en el tráfico de red. Esto se debe a que la CWT se considera un método eficaz para la extracción de características. Los resultados experimentales ponen de relieve que el uso de CWT con modelos de aprendizaje automático mejora el rendimiento de clasificación de MNIDS hasta un 3,36 % en precisión global. Además, mejora el valor F1-score hasta en un 40 % del total de clases utilizando el modelo propuesto.

Palabras clave: Aprendizaje automático; Deep Learning; Sistema de detección de intrusos; Ciberataques; Transformada Wavelet Continua.

INTRODUCTION

The fast advancement of technologies has raised the number of devices that are connected to the internet. There are many emergent technologies commonly used which focus on connecting devices to the internet, such as the Internet of Things (IoT).[1] The number of connected devices is expected to reach approximately 80 billion by 2030.[2] The fast increase in the number of connected devices every year results in a large amount of produced data that keeps growing rapidly. It also raises the complexity and the challenges in dealing with a huge amount of data,[3] different characteristic modes,[4] heterogeneous data,[5] and faster travel rates.[6] The IoT and other devices connected to the internet are vulnerable to different types of cyberattacks due to the complexity of the modern networks.[7] For example, compromised devices in a network controlled by an attacker who is responsible for bombarding a website that is cloud based by sending huge requests. This huge and excessive traffic results in overwhelming the cloud-based service, making access for authorized users difficult. This attack, called a Distributed Denial of Service (DDoS) attack, targets the network layer, and is considered as an extension of DoS attacks.[8]

Cybersecurity threats can vary from ransomware attacks that damage civil systems to spying on essential secret information to intelligent threats.[9] Traditional computer networks face challenges including the number of threats and the diversity of the threats.[10] As threats evolve, the techniques and mechanisms that detect them should counteract them. The National Institute of Standards and Technology (NIST) defines IDS as the act of scanning and analyzing network segments to capture attacks.[11] IDS can be hardware or software that can be used for monitoring procedures to protect the system from harmful activities.[12] It monitors network traffic to capture and alert administrators about possible unauthorized and malicious activity. This helps protect the violations of security protocols, such as data encryption, secure sockets layer (SSL) authentication, and firewall port settings.[13]

IDS can be categorized into three groups based on their installation in a system: hybrid IDS, host-based IDS (HIDS), and network-based IDS (NIDS). HIDS analyzes the critical operating system files and detects threats from a single computer system.[14] This approach often easily identifies attacks on that system, however some filtered malware could be extremely difficult to detect. A network intrusion detection system (NIDS) can be installed on a router or switch in a network. It uses different computer connections to identify malicious data.[14] NIDS is usually placed on the trusted side of network architecture. It scans through incoming network traffic to identify malicious activity. On the other hand, hybrid IDS may be installed both on hosts and on the network.[13] The major aim of NIDS is to detect malicious logging data and notify the manager of the network. NIDS does not prevent the system from intrusion attacks but generates alarms after detecting an attack in real-time or before it arrives. It is crucial to notify the system after an attack, as an IDS has the ability to maintain and update the profile of an intrusion inside the log. The operating system needs to manage disk space and CPU resources for log analysis. Managing log formats and comparing them with identified attack patterns is a significant challenge in IDS.[15]

Anomaly-based intrusion detection system (AIDS) and signature-based intrusion detection system (SIDS) are two types of detection techniques used to identify malicious data. SIDS is a detection method designed based on a signature for detecting recognized patterns of an attack.[16] AIDS is a detection method that is also called behavior-based IDS. It works by creating a profile for normal behavior in a system. Then, it compares this profile with any unusual activities, which helps to identify threats.[16] The primary advantages of AIDS are that it can detect new and unidentified threats and that its normal activity profile is customized for certain applications and networks. HIDS is a detection method that combines AIDS and SIDS.[17] Existing IDSs concentrate on binary classification and using traditional machine learning (ML) algorithms, focusing on classifying actions as malicious or normal without giving full attention to the details of the classification scheme, such as detecting the type of the attacks and distinguishing the risk level of an intrusion.[18] There are several issues of NIDS, including high false positive alarms and the constraint of adaptation when facing new threats.[13] The classification accuracy of ML models relies on different aspects such as the quality of training and testing data, feature selection, tuning the parameters of classifiers, and feature engineering.[19] The quality of data can be increased by performing data preprocessing steps. Feature selection is a technique that can be applied to selecting the most relevant and important features of a dataset to accomplish better results with reduced computation overhead. However, this technique can cause redundancy of features that provide similar information, especially with highly correlated features.[20] Feature engineering involves processing the original data to create new features to improve the performance of ML classifiers.[21] Feature engineering can overcome the issue of correlation complexity of features in data.

It is insufficient to depend only on ML classifiers to achieve higher classification accuracy, especially with dataset containing multiple classes. Employing techniques such as feature engineering with ML classifiers can enhance the classification accuracy of classifiers.

There is a need to categorize cyberattacks in an effective way. The challenge arises when there are a diverse number of classes, the prediction process becomes more complex, which negatively affects the classification accuracy.[22] Traditional NIDS classification approaches might not be the most effective method, this means that enhancing multiclass NIDS is required to deal with many classes. Therefore, using an efficient technique like continuous wavelet transform (CWT) can help in improving the classification accuracy of multiclass NIDS. In this paper, we exploit the power of integrating CWT with machine learning models to enhance the accuracy of multiclass network intrusion detection systems (MNIDS) when predicting cyberattacks in network traffic. The contribution of this paper is summarized as follows:

- Applying the continuous wavelet transform with an IDS dataset to extract features by computing the magnitude and phase values of each feature in network traffic. Then the computed values are added as new features to the IDS dataset.
- Implementing MNIDS using various classifiers of ML to analyze the effectiveness of integrating CWT.
- Conducting a comparative and analytical study of the classification performance of each ML classifier to observe their effectiveness when utilizing CWT.

The rest of the paper is organized as follows. Section 2 presents related works in the field of intrusion detection systems. Section 3 details the research methods. Section 4 presents results, and section 5 discusses findings; Section 6 concludes with key findings, implications, and recommendations for future research.

Related Works

There has been significant progress made in the domain of IDSs between 2020 and 2023. This review explores various methods that were published during this time frame. The related work section addresses deep learning and machine learning techniques, sensor fusion, control systems for autonomous cars, and classical and statistical methods in the field of IDSs. Additionally, some new approaches are also explored, including pipeline leak detection. The aim is to present a thorough grasp of the state of the art in IDS.

The authors in [23] proposed the Autoregressive Integrated Moving Average (ARIMA) and Z-score to reduce the training phase and vehicle reliance. However, a long window size is necessary for a satisfactory outcome, which would lengthen the detection time. The authors in [24] proposed a unique intrusion detection system known as the Clock Offset-based Intrusion Detection System (COIDS) to identify unusual behaviors. The authors employed the cumulative sum technique to track every unusual departure in the clock offset and active learning to shape the typical clock behavior of Electronic Control Units (ECUs). The authors in [25] created a technique for anomaly detection using unsupervised deep learning. The authors learned a behavior pattern from typical sensor signals using an artificial neural network and a deep autoencoder, then compared it with observations of the vehicle based on the nominal behavior they had developed for vehicle anomaly detection.

The authors in [26] proposed a sensor fusion technique based on the line of lane data captured by an onboard camera built into smart cars. This method complemented the sensor information using a visual-aided strategy.

By combining data from an inertial measurement unit (IMU) in driverless cars with that from a worldwide navigation satellite system (GNSS), Xiong et al.[27] demonstrated an innovative approach to sensor processing. A GNSS and IMU fusion-based approach was proposed by Liu et al.[28] to overcome measurement signal delay and deal with inaccuracies brought on by GNSS's low sampling rate. Researchers in [29] evaluated the K-Nearest Neighbor (KNN) method for network anomaly detection and tested the efficiency of the proposed intrusion detection system utilizing the KDD CUP 99 dataset. A detection model based on multiple traditional ML classification techniques was suggested by Alqahtani et al.[30] However, earlier approaches used in the field of intrusion detection had ineffective classification performance, resulting in a higher false positive rate (FP) and a lower rate of detection (DR) in the identification (ID) system.

In [31], the authors combined chi-square and random forest to provide a technique that is mixed feature selection (FS) for detecting intrusions. They evaluated the effectiveness of features and the correlation between data attributes and labels. The researchers utilized the ANOVA F-test when they considerd the univariate feature selection technique.[32] Also, they utilized a Kalman filter for prediction and an automated method of machine learning. They utilized Bayesian optimization as the optimization method for the architecture of neural networks search to choose the most accurate design among architectures that are in a list. The authors in [33] propsoed a Deep-AE-based model for detecting anomaly to create a model that is effective for intrusion detection that utilizes the Restricted Boltzmann Machine (RBM).

Authors in [34] propsoed an enhanced workflow for feature selection in intrusion detection systems. They utilized power transformation, multi objective optimization and normalization in their model. They have achieved F1 score is 93,17 % on dataset called ISCX-IDS2012 dataset and on dataset called CIC-IDS2017 they achieved F1 score is 99,69 %. This indicates that a significant improvment in feature selection techniques in Intrusion detection systems. The researchers in [35] proposed an artificial intelligence-based intrusion detection system (AI-IDS) for real-time HTTP traffic. Their research shows that the algorithm can differentiate between complex attacks and patterns that aresafe traffic. In addition, it improves the signature-based network intrusion detection s and refines Snort rules refined. The study in [36] conducted experiments and stated that in network anomaly detection deep learning methods can be utilized. They applied a technique to the anomaly using flow identification established based on a deep neural network. An ensemble-based method for detecting network anomalies in IDS was presented by Imran et al.[37] This approach breaks anomalies into several classes by combining prediction and learning processes.

The authors in [38] examined the effectiveness of four popular classifiers for binary classification on the UNSW-NB15 dataset: Support Vector Machine (SVM), Random Forest, Naive Bayes, and Decision Tree. After converting classified information to attribute values using one-hot encoding, the researchers ran machine learning upon the whole feature set. The trials' outcomes showed that the researchers' accuracy on SVM, Naive Bayes, Random Forest, and Decision Tree were 79,59 %, 66 %, 76 %, and 78 %, respectively. Hybrid techniques have surfaced to overcome the constraints in the design of IDS feature selection. By combining the wrapping and filtering procedures, these strategies maximize the utility and efficacy of both methods while also enhancing computation to improve predictions. Utilizing the dataset UNSW-NB15, the authors performed a comparison analysis of ML models in [39]. Ghurab et al. performed a thorough examination of NIDS benchmark datasets.[40] In [41], a thorough evaluation of supervised classifiers for NIDS design, the J48 Consolidated classifier was shown to be the best option. High-quality datasets are essential to train ML-based NIDS, according to Sarhan et al.[42]

An innovative IDS using deep learning techniques, specifically the combination of LSTM and CNN, was proposed by Kanna and Santhi. As a result of their model's superior accuracy, low false positive rate, and competitive classification coefficients, IDS performance has significantly improved.[43] Payload embeddings, a technique that combines byte embedded data and a shallow neural network, were proposed in [44]. This method outperformed conventional intrusion detection algorithms, consistently achieving rates of accuracy ranging from 75 % to 99 % across many datasets. It demonstrates how embeddings may be used to effectively identify network intrusions. Acknowledging the increasing demand for standardized methods in contemporary datasets for network intrusion detection, a trustworthy method for detecting pipeline leaks through AE signals was presented by Ahmad et al.[45] CWT is utilized to create AE pictures that show time-frequency scales, the technique effectively captures leak-related data using high-energy representations. Extracting global and local characteristics so these scalograms are processed by an ANN and a CAE. Improvement of the precision and dependability of leak detection, these characteristics are integrated into a single feature vector. A testbed dataset that is based on industrialized pipeline utilized to show the high accuracy of a shallow artificial neural network (ANN) when classifying the pipeline leak status, fluid pressures and irrespective of breach sizes.

Xia et al.[46] presented an automobile sensors data processing technique that estimates the yaw misalignment of the IMU in the car. The method was based solely on integrating the IMU with the onboard sensor without the aid of any outside data. A sensor fusion framework based on car chassis sensors and GNSS was presented by Gao et al.[47] In order to prevent or lessen collisions, Alsuwian et al.[48] suggested a unique enhanced emergency braking system (EBS) that uses sensor fusion and is capable of independently recognizing insecure driving conditions and then activating the vehicle's braking system. A fuzzy neural network-based active fault-tolerant control (AFTC) method for autonomous cars was also introduced by Alsuwian et al.[49] By successfully detecting any anomaly in wheel acceleration, the AFTC can stop possible instability issues in CAVs before they even develop. The trials were carried out with the reference dataset KDD CUP'99 by Qazi et al.[50] In another paper, the authors, Qazi et al.[51] suggested a deep learning system for detecting network intrusions based on a convolutional neural network with one dimension (1D-CNN). For the experiments, the researchers utilized the benchmark CICIDS2017 dataset. An AdaBoost-based network intrusion detection as well as classification system was proposed by Ahmad et al.[52] The UNSW-NB 15 dataset was utilized by the authors to discover network anomalies. The results of the experiment demonstrated that the suggested technique could detect various types of network intrusions on networks of computers. A branch of machine learning called "deep learning" uses hidden layers to identify the characteristics of a deep network. These methods are more effective than conventional machine learning [53] because of their all-inclusive structure and ability to independently extract and understand the pertinent features of the dataset. DL has been more popular recently and is being used for intrusion detection; studies show that DL works better than conventional techniques. ML approaches are being used progressively to enhance anomaly-based NIDS accuracy and decrease false positives.[54,55]

In 2022, Farrukh et al.[56] presented Payload Byte. This adaptable instrument expedites the curation of datasets and creates a uniform groundwork for further investigations. It provides the capacity to identify and label different protocols, which makes it possible to convert data with labels into byte-wise vectors of features that are used for training machine learning models. All this research points to how important payload analysis is for detecting network intrusions. These studies greatly contribute to the ongoing development of NIDS by employing a variety of approaches, from cuttin- edge technologies to NLP techniques, and they offer insightful information for dealing with cybersecurity risks. Ho et al. provided a novel approach to intrusion detection that combines a vision transformer (ViT) classifier with a flow to image conversion algorithm. This process converts traffic from the network flows into a series of vectors, which are then encoded into a space of latent information and decoded into pictures. With binary classification, the score of the F1- score is 96,3 % on UNSW NB15 and 98,5 % on the CIC IDS2017 dataset; their trials notably provide significant results. Their approach achieves an F1 score of 96,4 % in multiclass classification.[57] A thorough analysis of NIDS methodologies was provided by Albasheer et al., who emphasized the need for sophisticated ML algorithms and alert correlation.[58] To identify pipeline leaks utilizing AE signals, the technique presented herein shows a hybrid methodology that integrates the characteristics retrieved from STFT and CWT. A study of the AE signals that capture both spectral and temporal information is made possible by the integration of STFT and CWT, in contrast to other approaches that only use one transform technique. The time domain AE signals are converted into time frequency representations via STFT. Consequently, at various time intervals, the spectral content and energy distribution are detected.[59] In the realm of autonomous vehicles, Xia et al.[60] developed a unique sensor fusion strategy by combining GNSS-IMU fusion-based techniques with the vehicle dynamic model. In resource-constrained contexts, Rizvi presented a deep learning solution for intrusion detection that achieved high accuracy.[61] Table 1 presents a summary of related works in intrusion detection systems.

Table 1. Summary of related works

Reference	Description	Methodology
Tomlinson et al.[23]	Automotive CAN cyberattacks detection to identify anomalies of packet timing in time windows.	ARIMA and Z-score.
Halder et al.[24]	Proposed an IDS called COIDS based on clock offset.	Utilized active learning and cumulative sum to monitor anomalous clock offset.
He et al.[25]	Designed anomaly detection method based on an unsupervised deep learning.	Used ANN and deep autoencoder to learn normal behavior patterns from sensor messages.
Liu et al.[26]	Proposed a method in sensor fusion using visual-aided strategy with lane line data.	Complemented sensor information using lane line data from onboard camera.
Xiong et al.[27]	Proposed a novel method for sensor processing through fusing GNSS and IMU in automated vehicles.	Fused GNSS and IMU information for automated vehicles.
Liu et al.[28]	Presented a GNSS and IMU fusion-based method to overcome signal delay and address low sampling rate errors.	Fused GNSS and IMU information to address signal measurement issues.
Xu, H. et al.[29]	K-Nearest Neighbor for network anomalies using KDDCUP dataset.	K-Nearest Neighbor (KNN).
Alqahtani, H. et al.[20]	Cyber intrusion detection using ML classification techniques.	Multiple ML classification algorithms.
Song, J. et al.[31]	Hybrid feature selection for lightweight IDS.	Chi-square with RF.
Biney, G. et al.[32]	Adaptive scheme for ANOVA models.	ANOVA F-test.
Khan, M.A. et al.[33]	Efficient Conv-AE-Based IDS using heterogeneous dataset.	Convolutional Auto-Encoder (Conv-AE).
Siddiqi and Pak.[34]	Optimized feature selection process for IDS with normalization and multi-objective optimization.	Feature selection optimization.
Kim et al.[35]	AI-based IDS for real-time HTTP traffic distinguishing complex attacks from benign traffic patterns.	Deep learning for real-time HTTP traffic analysis.
Girdler, T. et al.[36]	Developed an ID and prevention system utilizing SDN.	Software-Defined Networking (SDN).
Imran, R. et al.[37]	Presents a group of learning and prediction methods to improve the accuracy of anomaly detection.	ANOVA F-test, automated ML, Kalman filter.

Hossain, Z. et al.(38)	NIDS using ML approaches.	SVM, RF, Naive Bayes, Decision Tree.
Disha and Waheed.(39)	Compared ML models by utilizing the dataset UNSW-NB15.	Machine learning model comparison.
Ghurab et al.(40)	Benchmark datasets for NIDS are analyzed in detail.	Dataset analysis.
Panigrahi et al.(41)	J48 consolidated classifier was found to be the ideal selection when supervised classifiers were evaluated for NIDS.	Supervised classifier assessment.
Sarhan et al.(42)	Shown how crucial high-quality datasets are to the training of ML-based NIDS.	Dataset quality analysis.
Kanna and Santhi.(43)	Proposed an inventive IDS harnessing CNNs and LSTMs for high accuracy and low false positive rate.	CNNs and LSTMs integration.
Hassan et al.(44)	Proposed payload embeddings for intrusion detection with byte embeddings and a shallow neural network.	Payload embeddings, shallow neural network.
Ahmad et al.(45)	Pipeline leak detection using AE signals and deep learning with acoustic imaging and CWT.	Application of CWT to generate AE images processed through CAE and ANN for feature extraction.
Xia et al.(46)	Proposed a vehicle sensor data processing method integrating onboard sensor with IMU.	Estimated yaw misalignment of IMU without external information.
Gao et al.(47)	Proposed a sensor fusion framework is proposed that is based on GNSS and vehicle chassis sensors.	Enhanced vehicle localization with the use of lateral velocity and onboard sensors.
Alsuwian et al.(48)	Proposed an advanced emergency braking system (EBS) using sensor fusion.	Autonomous detection of insecure driving states and triggering braking system.
Alsuwian et al.(49)	Proposed a fuzzy neural network-based active fault-tolerant control (AFTC) method.	Detected wheel speed abnormalities to prevent instability problems.
Qazi et al.(50)	Intelligent and efficient network IDS using deep learning.	Non-symmetric deep auto-encoder.
Qazi et al.(51)	In NIDS, a deep learning system based on 1D-CNN is proposed.	1D-Convolutional Neural Network.
Ahmad, I. et al.(52)	Efficient NIDS and classification system based on AdaBoost is presented.	AdaBoost approach.
Ahmed et al.(54)	Analyzed ML/DL solutions for network threat detection in SDN-based platforms.	State-of-the-art ML/DL solutions analysis.
Farrukh et al.(56)	Introduced Payload-Byte tool for standardized dataset curation and feature extraction.	Payload-Byte tool for dataset curation and feature extraction.
Ho et al.(57)	Innovative IDS using flow to image conversion and vision transformer classifier.	Flow-to-image conversion, vision transformer classifier.
Albasheer et al.(28)	A survey of NIDS approaches, highlighting alert correlation and advanced ML.	Survey of NIDS approaches.
Jiang et al.(59)	Hybrid approach for pipeline leak detection using AE signals with STFT and CWT.	Combining CWT with STFT for a more analysis of AE signals.
Xia et al.(60)	Sensor fusion method introduced to integrate GNSS-IMU with vehicle dynamic model.	Synthesized kinematics and dynamics of autonomous vehicle for sideslip angle estimation.
Rizvi et al.(61)	Deep learning method used for intrusion detection in resource-constrained environments with high accuracy.	Deep learning for resource-constrained environments.

Most of the related works are aligned with the contribution of this paper, since most of the researchers have made efforts to enhance the accuracy of intrusion detection by using various methods. However, improving the accuracy of intrusion detection systems is still an important aspect in the field, especially with the increase in the number of cyberattack categories. This is considered challenging for conventional IDS to have high classification accuracy with great number of cyberattack types. Therefore, this paper addresses this issue by integrating the CWT with ML model to have high classification accuracy for multiclass cyberattacks.

METHOD

This section explores and discusses the research methodology of this study. First, it demonstrates the operational environment of the proposed MNIDS. Second, it explores the dataset utilized to assess and evaluate the performance of the proposed MNIDS. Third, it explains the theory of the CWT. Fourth, it discusses classifiers based on machine learning and deep learning that are used to implement MNIDS.

The Architecture of the Proposed Model

Figure 1 demonstrates the operation environment of the proposed model. The network traffic coming from the internet is usually untrusted traffic and potentially has malicious activity that could pass firewalls.

Therefore, it is important to not be limited to only detecting cyberattacks but also to learn the attack types. The proposed model detects network traffic to identify and categorize cyberattacks based on their types. It uses CWT and MNIDS techniques. The CWT is integrated to enhance the classification accuracy of the MNIDS. It is applied to the incoming network traffic and then its results are added to the features of the network traffic. The main goal of using CWT is to improve the detection accuracy of MNIDS. Finally, the MNIDS classifies cyberattacks into categories.

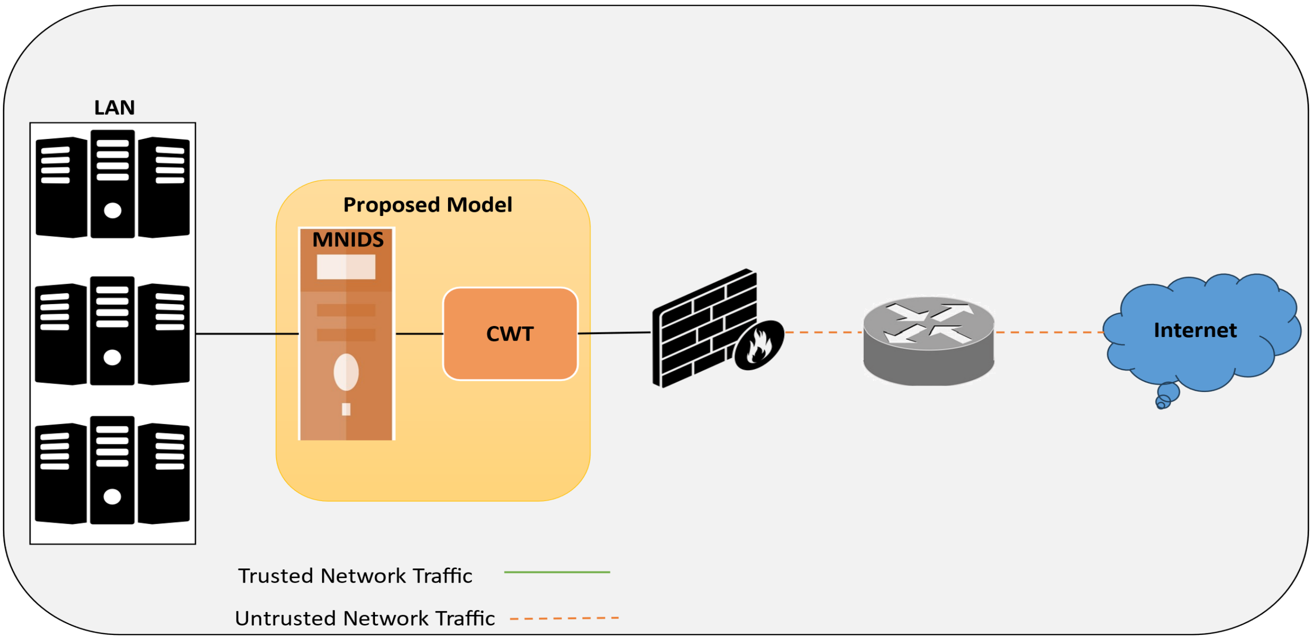


Figure 1. The Proposed Model and the Operational Environment

Dataset

This research leverages the CSE-CIC-IDS2018 (62) dataset because it contains fourteen cyberattack types. It is considered as an IDS dataset of network traffic that was built on Amazon Web Services (AWS). In addition, it can be utilized to evaluate the detection mechanism and the ability of a classifier to predict the category of cyberattacks. CSE-CIC-IDS2018 is a popular dataset, and it is frequently utilized by researchers, this explains its use in this study. The dataset comprises eighty columns in total, seventy-nine columns are used as features of network traffic and the last column is used as the label of the cyberattack type. It consists of ten comma-separated values (CSV) files and each file holds numbers of rows as listed in Table 2.(62)

Reviewing the table, each file contains a few types of cyberattacks, nevertheless, to serve the purpose of this research we combined all types of cyberattacks which are provided in the dataset in a single CSV file. This objective is achieved by selecting a maximum of 2 000 rows of each cyberattack category from each CSV file as shown in table 3. It can be observed that some cyberattacks contain a number of rows less than 2 000 because the original dataset includes a number of rows less than 2 000 rows. As a result, there are 15 unique classes in the aggregated dataset. Class 0 refers to "Benign" which means normal network traffic. Classes 1-14 refer to the attack category. The total number of rows in the aggregated dataset is 413 648 and it includes fifteen categories of classes.

Table 2. Distribution of CSE-CIC-IDS2018 Dataset

CSV File Name	Label Name	Number of Row
02-14-2018	Benign	667 626
	FTP-BruteForce	193 360
	SSH-BruteForce	187 589
02-15-2018	Benign	996 077
	DoS attacks-GoldenEye	41 508
	DoS attacks-Slowloris	10 990
02-16-2018	Benign	446 772
	DoS attacks-Hulk	461 912
	DoS attacks-SlowHTTPTest	139 890

02-20-2018	Benign	7 372 557
	DDoS attacks-LOIC-HTTP	576 191
02-21-2018	Benign	360 833
	DDoS attack-HOIC	686 012
	DDoS attack-LOIC-UDP	1 730
02-22-2018	Benign	1048 213
	Brute Force -Web	249
	Brute Force -XSS	79
	SQL Injection	34
02-23-2018	Benign	1048 009
	Brute Force -Web	362
	Brute Force -XSS	151
	SQL Injection	53
02-28-2018	Benign	544 200
	Infiltration	68 871
03-01-2018	Benign	238 037
	Infiltration	93 063
03-02-2018	Benign	762 384
	Botnet	286 191
Total Rows		16 232 943

Table 3. Aggregated Cyberattack Dataset

Class #	**Cyberattacks**	**Number of Rows**
0	Benign	200 000
1	Botnet	20 000
2	Brute Force -Web	611
3	Brute Force -XSS	230
4	DDoS attack-HOIC	20 000
5	DDoS attack-LOIC-UDP	1 730
6	DDoS attacks-LOIC-HTTP	20 000
7	DoS attacks-GoldenEye	20 000
8	DoS attacks-Hulk	20 000
9	DoS attacks-SlowHTTPTest	20 000
10	DoS attacks-Slowloris	10 990
11	FTP-BruteForce	20 000
12	Infiltration	40 000
13	SQL Injection	87
14	SSH-BruteForce	20 000
Total Rows		413 648

Continuous Wavelet Transform (CWT)

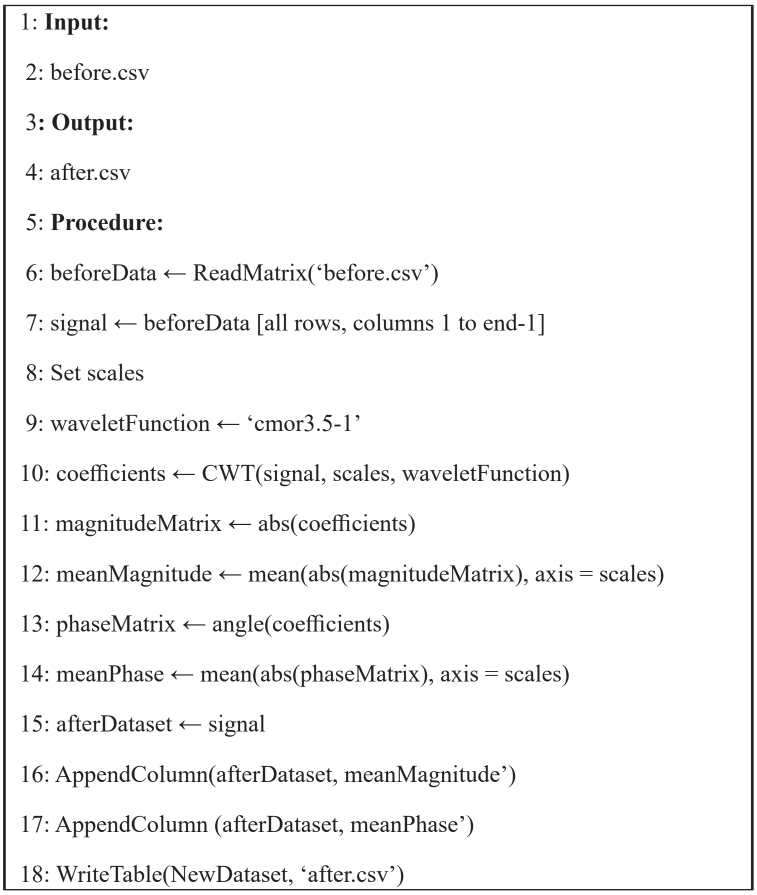
CWT is a mathematical technique that is used by researchers in different domains such as signal processing and image processing.[64,65] In signal processing, CWT is an effective tool to reduce noise level and extract features for further analysis. In image processing, it is applicable for image compression and texture analysis. CWT breaks signals into small waves to analyze and provide information from frequency domain and time domain. As opposed to Fourier transform which focuses only on frequency domain of a signal.

CWT can be expressed mathematically using equation 1.[65] Where W is the desired output in the CWT. is the width of wavelet and refers to the translation factor. x*(t)* refers to the original signal. 1/√a represents the energy of the wavelet and is the analytical function.

$$W(a,b) = \int_{-\infty}^{\infty} x(t) \cdot \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) dt \quad (1)$$

Algorithm 1 demonstrates how CWT is applied in this study. Algorithm 1 is constructed using three parameters Input, Output, and Procedure. The Input contains the dataset before implementing CWT. The dataset includes seventy-nine features and column number eighty is used for labelling. The baseline dataset is named *before.csv* and the total number of rows in its file is 413 648. The Output holds the generated *after.csv* file after implementing CWT. It includes eighty-one features, the previous seventy-nine features plus two features generated by the algorithm, and the last column used for labeling. The total number of rows in the *after.csv* file is 413 648.

The Procedure performs the processing of Algorithm 1. Initially, the variables *beforeData* and signal are initialized. *beforeData* is used to store the data of the *before.csv* file, which is converted from CSV format to matrix. To exclude the column that is used for labeling, all features except the last column are stored in the variable signal. After that, the wavelet parameters are defined. The *scales* variable is used to set wavelet scale, and the wavelet function used to specify the use of Complex Morlet wavelet function, which is set to 'cmor3.5-1'. Then the CWT is performed on the signal and the results are stored in the coefficients variable as complex numbers. Each complex number has magnitude and phase. The magnitude of wavelet measures the strength of the wavelet coefficients is stored in *magnitudeMatrix*. Then the mean of magnitude is calculated to extract meaningful summary of the processed data. The phase analyzes the shift and the change in the signal's behavior over time. It is stored in *phaseMatrix*. A summary of the phase is stored in *meanPhase*. Finally, the *meanMagnitude* and *meanPhase* are appended to the *afterDataset*, and then it is saved in the *after.csv* file.



```
Input:
before.csv
Output:
after.csv
Procedure:
beforeData ← ReadMatrix('before.csv')
signal ← beforeData [all rows, columns 1 to end-1]
Set scales
waveletFunction ← 'cmor3.5-1'
coefficients ← CWT(signal, scales, waveletFunction)
magnitudeMatrix ← abs(coefficients)
meanMagnitude ← mean(abs(magnitudeMatrix), axis = scales)
phaseMatrix ← angle(coefficients)
meanPhase ← mean(abs(phaseMatrix), axis = scales)
afterDataset ← signal
AppendColumn(afterDataset, meanMagnitude')
AppendColumn (afterDataset, meanPhase')
WriteTable(NewDataset, 'after.csv')
```

Figure 3. Algorithm 1 Apply CWT

Classifiers

To explore the effectiveness of using CWT in improving the classification performance of MNIDS, we implemented various classifiers named, Random Forest (RF), Decision Tree (DT), Feedforward Neural Network (FFNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU).

RF is an ensemble classifier comprised of sets of decision trees and each set votes to the final decision. Each tree is built with random groups of features and trained data to reduce overfitting.(66) RF classifier was trained with standard parameters. There are 100 trees, and Gini is used for evaluating the quality of a split.

The DT classifier is simpler and faster than the RF classifier. In addition, it is suitable to deal with high-dimensional data efficiently; for that reason, DT is employed in data engineering to analyze various types of data.(67) DT classifier was trained with standard parameters.

The FFNN classifier is designed based on neural network. It usually consists of an input layer, hidden layer, and output layer. The input layer receives the input data as vector. The hidden layer processes the input data, and the output layer provides the decision made by the neural network.(68) The FFNN classifier was trained with 100 neurons in the hidden layer, with the use of ReLU activation function, and Adam is used as a weight optimization.

The LSTM classifier is considered as an advanced recurrent neural network. LSTM can remember information for longer duration which is useful with sequential data. It comprises three gates, which are forget gate, input gate, and output gate. Forget gate decides whether the incoming information is important or not, therefore important information is held and unimportant information is forgotten. The important information is written in the LSTM's memory through the input gate after deciding what new information can be added to the cell. The output gate controls which cell to be used when predicting.[69] The LSTM classifier was trained with 50 units, Adam was used as a weight optimization, the epoch was equal to 50, and the batch size was equal to 32.

The GRU classifier simplifies the LSTM model by combining the forget gate and the input gate in one gate named update gate. In addition, GRU has a second gate called rest gate used to manage data inside the unit unlike the LSTM which requires a separate memory cell.[70] The GRU classifier was trained with Adam as a weight optimization, the epoch was equal to 50 and the batch size was equal to 32.

System Flows

Figure 3 illustrates the flowchart of the proposed model. Initially, the before.csv file was imported into MATLAB project. It contains a collection of network traffic data with malicious activity as discussed previously. Then the CWT is computed which generates the magnitude and the phase values. Both values are added as new features to the dataset. The CWT is calculated using MATLAB's built-in function called cwt.[71] The new dataset is saved as after.csv that contains network traffic data, magnitude, and phase values. Then the data in after.csv file is split into two sets as 80:20 ratio, 80 % is used for training and 20 % is used for testing. The train set is used to train RF, DT, FFNN, LSTM, and GRU classifiers. The test set is used to evaluate each classifier by calculating F1-score and test accuracy as shown in equations 2-3.[72] In addition, the percentage increase in test accuracy is calculated using equation 4. F1-score measures a classifier performance after combining precision and recall as shown in equation 2. Therefore, it provides a single accuracy performance metric that is useful for imbalanced classes. Test accuracy measures the overall accuracy of a classifier as shown in equation 3. Basically, it evaluates all correct predictions made by a classifier out of all the number of predictions. The percentage increase is computed as shown in equation 4. It indicates the percentage enhancement of the test accuracy of a classifier.

$$F_1 \text{ score} = 2 * \left(\frac{Precision \times Recall}{Precision + Recall}\right) \times 100 \quad (2)$$

$$\text{Test Accuracy} = \left(\frac{True\ Positive + True\ Negative}{True\ Positive + False\ Positive + True\ Negative + False\ Negative}\right) \times 100 \quad (3)$$

$$\text{Percentage Increase} = \left(\frac{New\ Value - Old\ Value}{Old\ Value}\right) \times 100 \quad (4)$$

The reason for evaluating each classifier with the above equations is to assess its classification performance. In addition, it is to compare the results of baseline classifiers (when the proposed model is not used) with classifiers that used the proposed model.

RESULT

This section presents and discusses the findings of the experimental results. As mentioned in the previous section the CSE-CIC-IDS2018 dataset was used to evaluate the classification accuracy of the proposed model.

Table 4 presents the accuracy performance, measured by F1-score for the baseline classifiers using RF, DT, FFNN, LSTM, and GRU classifiers. In this case baseline means without utilizing the proposed model. Meanwhile, Table 5 shows the F1-score results for the same ML models but with using the proposed model. Basically, each classifier is evaluated twice, without utilizing the proposed model and with utilizing the proposed model. Each table contains fifteen classes in total, fourteen of them (class 1-14) represent network traffic related to cyberattacks while class number 0 represents normal traffic.

For the RF classifier, F1-score was improved by the proposed model, classes 0,2,9,11, and 12 indicated a rise in the F1-score value. In addition, the results of the F1-score for the DT classifier show that DT gained an increase in the attack classes 3,5,9,11,12, and 13. Class 9 had a significant increase in F1-score. Regarding the F1-score values of the FFNN classifier, the results were enhanced in classes 2,3,9, and 11 using the proposed model. Class 2 increases by 3,03 %, class 3 improves by 29,31 %, class 9 enhances by 19,74 %, and class 11 increases by 56,90 %.

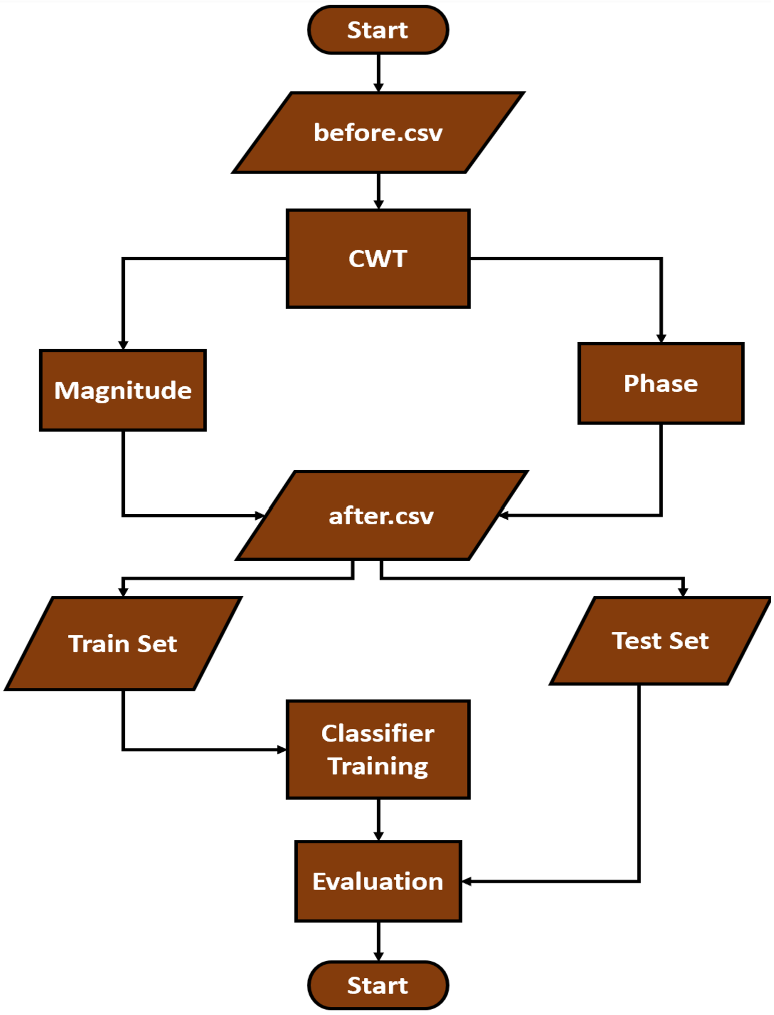


Figure 3. Flowchart of Proposed Model

The F1-score value for the LSTM classifier was improved in classes 0,9,11 and 12. The performance of F1-score of classes 0, and 9 was enhanced by 1,02 %. Meanwhile class 11 improved by 2,06 % and class 12 increased by 2,20 %. Finally, classes 2, 9,11,12, and 13 experienced an increase in the F1-score value of the GRU classifier. Thus, class 2 increased by 1,01 %, classes 9 and 11 enhanced by 1,02 %, class 12 increased by 1,11 %, and class 13 improved by 6,38 %.

Table 4. F1-Score Results of Baseline Models

Class #	RF	DT	FFNN	LSTM	GRU
0	90 %	89 %	92 %	98 %	98 %
1	100 %	100 %	100 %	100 %	100 %
2	83 %	89 %	66 %	99 %	99 %
3	92 %	95 %	58 %	99 %	100 %
4	100 %	100 %	100 %	100 %	100 %
5	100 %	99 %	100 %	100 %	100 %
6	100 %	100 %	100 %	100 %	100 %
7	100 %	100 %	100 %	100 %	100 %
8	100 %	100 %	100 %	100 %	100 %
9	63 %	63 %	76 %	98 %	98 %
10	100 %	100 %	100 %	100 %	100 %
11	74 %	74 %	58 %	97 %	98 %
12	41 %	42 %	38 %	91 %	90 %
13	83 %	69 %	73 %	94 %	94 %
14	100 %	100 %	100 %	100 %	100 %

Table 5. F1-Score Results of Proposed Models

Class #	RF	DT	FFNN	LSTM	GRU
0	91 %	89 %	92 %	99 %	98 %
1	100 %	100 %	100 %	100 %	100 %
2	87 %	89 %	68 %	99 %	100 %
3	92 %	96 %	75 %	99 %	100 %
4	100 %	100 %	100 %	100 %	100 %
5	100 %	100 %	99 %	100 %	100 %
6	100 %	100 %	100 %	100 %	100 %
7	100 %	100 %	100 %	100 %	100 %
8	100 %	100 %	100 %	100 %	100 %
9	92 %	90 %	91 %	99 %	99 %
10	100 %	100 %	100 %	100 %	100 %
11	92 %	90 %	91 %	99 %	99 %
12	46 %	46 %	34 %	93 %	91 %
13	73 %	70 %	44 %	94 %	100 %
14	100 %	100 %	100 %	100 %	100 %

DISCUSSION

This section discusses the performance of the proposed model against the baseline model of each ML classifier independently to further analysis the effectiveness of using CWT with ML classifiers. Figure 4 compares the F1-score performance of the baseline model and the proposed model of RF classifier. The x-axis represents the classes of the dataset, and the y-axis represents the performance of the F1-score for both models. The orange bars refer to F1-score without using the proposed model and the green bars refer to the F1-score when using the proposed model. The improvement of the F1-score of RF classifiers is clearly noticed in classes 9 and 11. Meanwhile, there is a slight increase in the performance of RF classifiers in classes 0, 2, and 12. Therefore as five classes improved using the proposed model with RF classifier, it can be observed that 33,33 % of the total classes were enhanced.

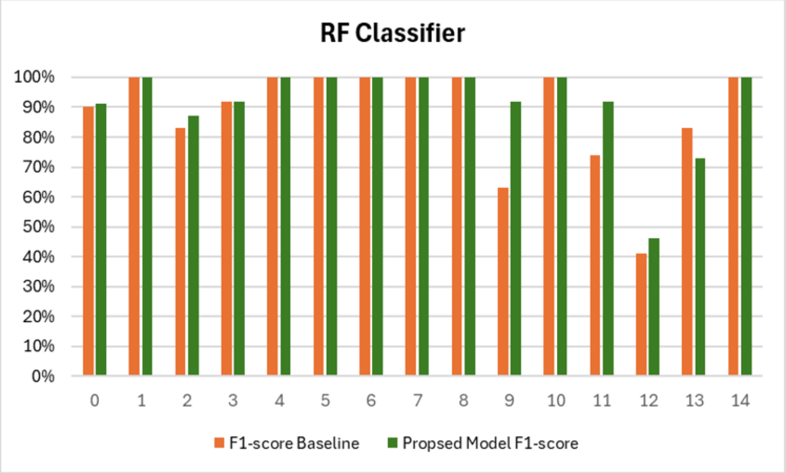


Figure 4. RF Classifier F1-score: baseline vs. proposed model

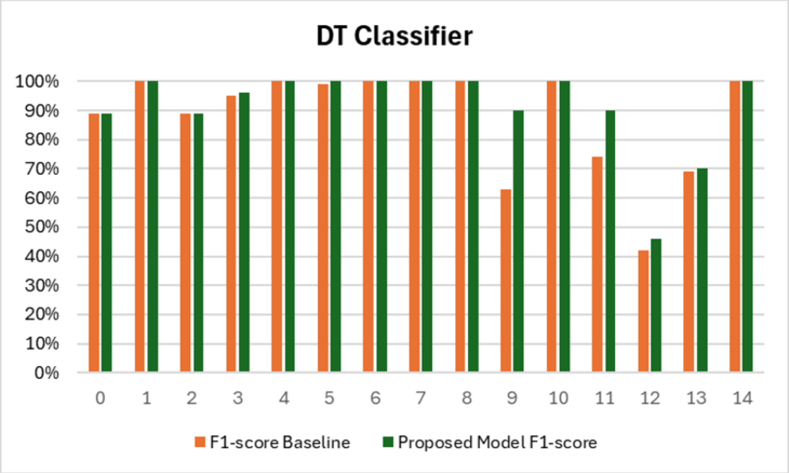


Figure 5. DT Classifier F1-score: baseline vs. proposed model

The performance of F1-score of DT classifier using and not using the proposed model is depicted in figure 5. Classes 9 and 11 indicate a significant rise in performance using the proposed model. At the same time classes 3,5,12 and 13 present a small increase in the F1-score performance. It can be noticed that six classes were enhanced using the proposed model with DT classifier, therefore 40 % of the classes improved, as the figure shows.

Figure 6 illustrates a comparison study of the FFNN classifier performance when the proposed model is not in used and when it is used. The performance is measured using F1-score. Classes 3,9, and 11 demonstrate an observable increase, while class 2 shows a minor increase. Therefore, 26,66 % of the total classes improved as indicated in the figure.

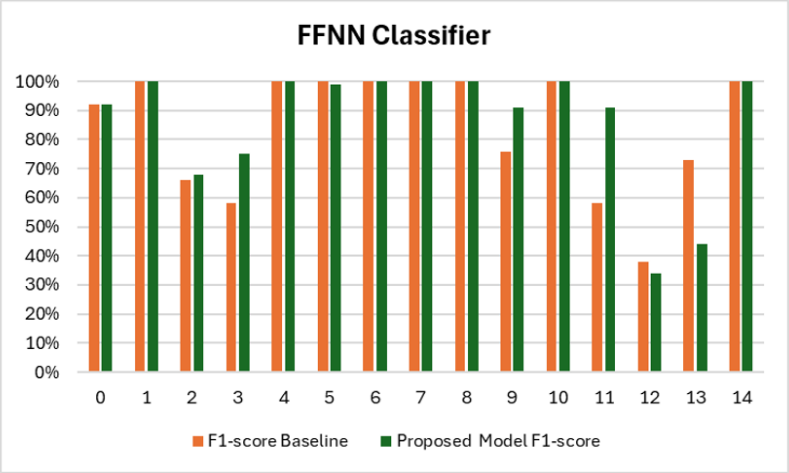


Figure 6. FFNN Classifier F1-score: baseline vs. proposed model

Figure 7 compares the F1-score performance of LSTM classifier using the proposed model and without using the proposed model. The improvement of F1-score of RF classifier is clearly noticed in classes 0, 9, 11, and 12. Therefore, four classes improved using the proposed model using LSTM classifier, as a result 26,66 % of the total classes being enhanced.

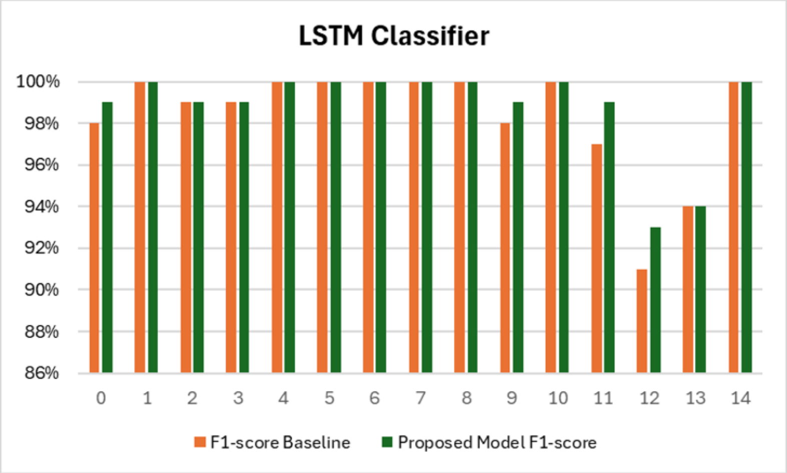


Figure 7. LSTM Classifier F1-score: baseline vs. proposed model

The performance of F1-score of GRU classifier using and not using the proposed model is depicted in figure 8. Class 13 shows a fast improvement of F1-score using the proposed model. Meanwhile classes 2, 9,11 and 12 indicate a rise in performance using the proposed model. It can be observed that five classes were enhanced using the proposed model with GRU classifier, therefore 33,33 % of the classes improved, as the figure shows.

Table 6 summarizes the overall testing accuracy of each classifier utilized in this study. The improvement in percentage when employing the proposed model ranges between 0,40 % to 3,32 %. RF classifier using the proposed model shows the best enhancement compared with its baseline result which increased from 87,35 % to 90,25 % that enhanced by 3,32 %. However, LSTM classifier demonstrates the smallest improvement in accuracy with using the proposed model compared with its baseline result which increased from 98,11 % to 98,50 % by 0,40 %. The reason for the slight increase in LSTM and GRU classifiers compared with RF, DT, and

FFNN is because LSTM and GRU both have high test accuracy baselines. To sum up, LSTM classifier presents the highest classification accuracy among other classifiers, however RF classifier indicates the highest percentage of improvement compared to its baseline outcome.

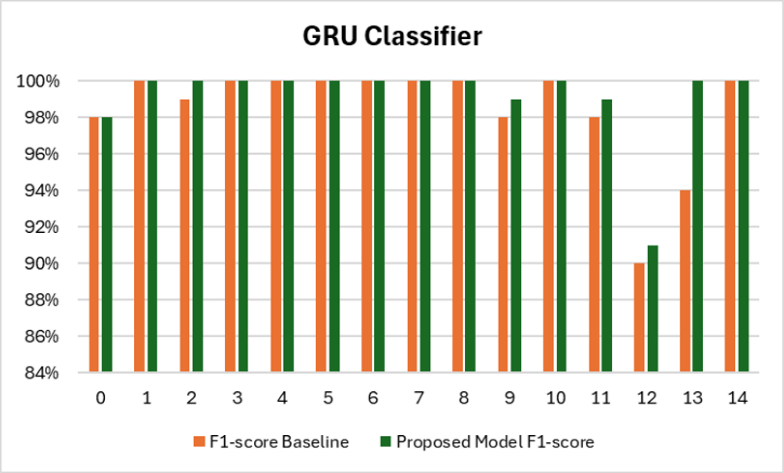


Figure 8. GRU Classifier F1-score: baseline vs. proposed model

Table 6. Percentage of Classifier Improvement

Classifier name	Test Accuracy Baseline	Test Accuracy with Proposed Model	Percentage of Increase
RF	87,35 %	90,25 %	3,32 %
DT	86,04 %	88,62 %	3,00 %
FFNN	88,44 %	90,75 %	2,61 %
LSTM	98,11 %	98,50 %	0,40 %
GRU	97,73 %	98,24 %	0,52 %

CONCLUSIONS

In conclusion, network traffic is susceptible to cyberattacks which require not only predicting the presence of cyberattacks, but also to classify their types. Implementing an accurate multiclass IDS using a machine learning approach can be crucial, especially with a diverse range of cyberattack types. Due to the fact that wide array of cyberattack types impact the classification accuracy. Merely training classifiers is not enough for obtaining higher accuracy. To overcome the above-mentioned challenge, this paper proposed the MNIDS that was implemented using machine learning with the integration of CWT to increase classification accuracy. CWT was applied with various types of machine learning classifiers such as RF, DT, FFNN, LSTM, and GRU to explore and analyze the effectiveness of CWT. The experimental findings indicate that employing CWT with a classifier boosts prediction accuracy of the multiclass classifier in classifying cyberattacks. Moreover, CWT enhances the overall accuracy of classifiers as the percentage of overall accuracy improved between 0,40 % to 3,32 %. The improvement of F1-score value ranges between 26,66 % and 40 % of the total classes using the proposed model. Therefore, for future work we recommend studying the computation process of CWT to envision the possibility of applying it with multiclass IDS in the IoT domain which will be a challenge because of the capability limitation of IoT devices.

REFERENCES

1. Ahmed SF, Alam MdS Bin, Afrin S, Rafa SJ, Taher SB, Kabir M, et al. Toward a Secure 5G-Enabled Internet of Things: A Survey on Requirements, Privacy, Security, Challenges, and Opportunities. IEEE Access. 2024;12:13125-45.

2. Chettri L, Bera R. A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems. IEEE Internet Things J. 2020 Jan;7(1):16–32.

3. Tran T, Navratil D, Sanders P, Hart J, Odarchenko R, Barjau C, et al. Enabling Multicast and Broadcast in the 5G Core for Converged Fixed and Mobile Networks. IEEE Transactions on Broadcasting. 2020 Jun;66(2):428–39.

4. Kumar N, Khanna R. A compact multi-band multi-input multi-output antenna for 4G/5G and IoT devices

using theory of characteristic modes. International Journal of RF and Microwave Computer-Aided Engineering. 2020 Jan 21;30(1).

5. Asad M, Basit A, Qaisar S, Ali M. Beyond 5G: Hybrid End-to-End Quality of Service Provisioning in Heterogeneous IoT Networks. IEEE Access. 2020;8:192320–38.

6. Dhasarathan V, Singh M, Malhotra J. Development of high-speed FSO transmission link for the implementation of 5G and Internet of Things. Wireless Networks. 2020 May 15;26(4):2403–12.

7. Liyakat KKS. Machine Learning Approach Using Artificial Neural Networks to Detect Malicious Nodes in IoT Networks. In 2024. p. 123–34.

8. Aslan Ö, Aktuğ SS, Ozkan-Okay M, Yilmaz AA, Akin E. A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions. Electronics (Basel). 2023 Mar 11;12(6):1333.

9. Chakraborty A, Biswas A, Khan AK. Artificial Intelligence for Cybersecurity: Threats, Attacks and Mitigation. 2022 Sep 27;

10. Bringhenti D, Marchetto G, Sisto R, Valenza F. Automation for Network Security Configuration: State of the Art and Research Trends. ACM Comput Surv. 2024 Mar 31;56(3):1–37.

11. National Institute of Standards and Technology. Intrusion Detection System [Internet]. 2024 [cited 2024 May 18]. Available from: https://csrc.nist.gov/glossary/term/intrusion_detection_system

12. NIST. https://csrc.nist.gov/glossary/term/intrusion_detection_system. 2022. Committee on National Security Systems (CNSS) Glossary.

13. Kumar S, Gupta S, Arora S. Research Trends in Network-Based Intrusion Detection Systems: A Review. IEEE Access. 2021;9:157761–79.

14. Satilmiş H, Akleylek S, Tok ZY. A Systematic Literature Review on Host-Based Intrusion Detection Systems. IEEE Access. 2024;12:27237–66.

15. Liu H, Lang B. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. Applied Sciences. 2019 Oct 17;9(20):4396.

16. Düzgün B, **Çayır** A, Ünal U, Dağ H. Network intrusion detection system by learning jointly from tabular and text-based features. Expert Syst. 2024 Apr 12;41(4).

17. Azam Z, Islam MdM, Huda MN. Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree. IEEE Access. 2023;11:80348–91.

18. Dini P, Elhanashi A, Begni A, Saponara S, Zheng Q, Gasmi K. Overview on Intrusion Detection Systems Design Exploiting Machine Learning for Networking Cybersecurity. Applied Sciences. 2023 Jun 25;13(13):7507.

19. Ghosh P, Azam S, Jonkman M, Karim A, Shamrat FMJM, Ignatious E, et al. Efficient Prediction of Cardiovascular Disease Using Machine Learning Algorithms With Relief and LASSO Feature Selection Techniques. IEEE Access. 2021;9:19304–26.

20. Khaire UM, Dhanalakshmi R. Stability of feature selection algorithm: A review. Journal of King Saud University - Computer and Information Sciences. 2022 Apr;34(4):1060–73.

21. Li T, Kou G, Peng Y. Improving malicious URLs detection via feature engineering: Linear and nonlinear space transformation methods. Inf Syst. 2020 Jul;91:101494.

22. Acharya T, Khatri I, Annamalai A, Chouikha MF. Efficacy of Machine Learning-Based Classifiers for Binary and Multi-Class Network Intrusion Detection. In: 2021 IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS). IEEE; 2021. p. 402–7.

23. Tomlinson A, Bryans J, Shaikh SA, Kalutarage HK. Detection of Automotive CAN Cyber-Attacks by Identifying Packet Timing Anomalies in Time Windows. In: 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). IEEE; 2018. p. 231–8.

24. Halder S, Conti M, Das SK. COIDS. In: Proceedings of the 21st International Conference on Distributed Computing and Networking. New York, NY, USA: ACM; 2020. p. 1–10.

25. He T, Zhang L, Kong F, Salekin A. Exploring Inherent Sensor Redundancy for Automotive Anomaly Detection. In: 2020 57th ACM/IEEE Design Automation Conference (DAC). IEEE; 2020. p. 1–6.

26. Liu W, Xiong L, Xia X, Lu Y, Gao L, Song S. Vision-aided intelligent vehicle sideslip angle estimation based on a dynamic model. IET Intelligent Transport Systems. 2020 Oct 20;14(10):1183–9.

27. Xiong L, Xia X, Lu Y, Liu W, Gao L, Song S, et al. IMU-Based Automated Vehicle Body Sideslip Angle and Attitude Estimation Aided by GNSS Using Parallel Adaptive Kalman Filters. IEEE Trans Veh Technol. 2020 Oct;69(10):10668–80.

28. Liu W, Xia X, Xiong L, Lu Y, Gao L, Yu Z. Automated Vehicle Sideslip Angle Estimation Considering Signal Measurement Characteristic. IEEE Sens J. 2021 Oct 1;21(19):21675–87.

29. Xu H, Przystupa K, Fang C, Marciniak A, Kochan O, Beshley M. A Combination Strategy of Feature Selection Based on an Integrated Optimization Algorithm and Weighted K-Nearest Neighbor to Improve the Performance of Network Intrusion Detection. Electronics (Basel). 2020 Jul 27;9(8):1206.

30. Alqahtani H, Sarker IH, Kalim A, Minhaz Hossain SMd, Ikhlaq S, Hossain S. Cyber Intrusion Detection Using Machine Learning Classification Techniques. In 2020. p. 121–31.

31. Song J, Zhao W, Liu Q, Wang X. Hybrid feature selection for supporting lightweight intrusion detection systems. J Phys Conf Ser. 2017 Aug;887:012031.

32. Biney G, Okyere GA, Alhassan A. Adaptive Scheme for ANOVA Models. Journal of Advances in Mathematics and Computer Science. 2020 Jun 20;12–23.

33. Khan MA, Kim J. Toward Developing Efficient Conv-AE-Based Intrusion Detection System Using Heterogeneous Dataset. Electronics (Basel). 2020 Oct 26;9(11):1771.

34. Siddiqi MA, Pak W. Optimizing Filter-Based Feature Selection Method Flow for Intrusion Detection System. Electronics (Basel). 2020 Dec 10;9(12):2114.

35. Kim A, Park M, Lee DH. AI-IDS: Application of Deep Learning to Real-Time Web Intrusion Detection. IEEE Access. 2020;8:70245–61.

36. Girdler T, Vassilakis VG. Implementing an intrusion detection and prevention system using Software-Defined Networking: Defending against ARP spoofing attacks and Blacklisted MAC Addresses. Computers & Electrical Engineering. 2021 Mar;90:106990.

37. Imran, Jamil F, Kim D. An Ensemble of Prediction and Learning Mechanism for Improving Accuracy of Anomaly Detection in Network Intrusion Environments. Sustainability. 2021 Sep 8;13(18):10057.

38. Hossain Z, Rahman Sourov MdM, Khan M, Rahman P. Network Intrusion Detection using Machine Learning Approaches. In: 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). IEEE; 2021. p. 438–42.

39. Disha RA, Waheed S. A Comparative study of machine learning models for Network Intrusion Detection System using UNSW-NB 15 dataset. In: 2021 International Conference on Electronics, Communications and Information Technology (ICECIT). IEEE; 2021. p. 1–5.

40. Ghurab M, Gaphari G, Alshami F, Alshamy R, Othman S. A Detailed Analysis of Benchmark Datasets for Network Intrusion Detection System. Asian Journal of Research in Computer Science. 2021 Apr 14;14–33.

41. Panigrahi R, Borah S, Bhoi AK, Ijaz MF, Pramanik M, Jhaveri RH, et al. Performance Assessment of Supervised Classifiers for Designing Intrusion Detection Systems: A Comprehensive Review and Recommendations for Future Research. Mathematics. 2021 Mar 23;9(6):690.

42. Sarhan M, Layeghy S, Moustafa N, Portmann M. NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems. In 2021. p. 117–35.

43. Rajesh Kanna P, Santhi P. Unified Deep Learning approach for Efficient Intrusion Detection System using Integrated Spatial–Temporal Features. Knowl Based Syst. 2021 Aug;226:107132.

44. Hassan M, Haque ME, Tozal ME, Raghavan V, Agrawal R. Intrusion Detection Using Payload Embeddings. IEEE Access. 2022;10:4015–30.

45. Ahmad S, Ahmad Z, Kim CH, Kim JM. A Method for Pipeline Leak Detection Based on Acoustic Imaging and Deep Learning. Sensors. 2022 Feb 17;22(4):1562.

46. Xia X, Xiong L, Huang Y, Lu Y, Gao L, Xu N, et al. Estimation on IMU yaw misalignment by fusing information of automotive onboard sensors. Mech Syst Signal Process. 2022 Jan;162:107993.

47. Gao L, Xiong L, Xia X, Lu Y, Yu Z, Khajepour A. Improved Vehicle Localization Using On-Board Sensors and Vehicle Lateral Velocity. IEEE Sens J. 2022 Apr 1;22(7):6818–31.

48. Alsuwian T, Saeed RB, Amin AA. Autonomous Vehicle with Emergency Braking Algorithm Based on Multi-Sensor Fusion and Super Twisting Speed Controller. Applied Sciences. 2022 Aug 24;12(17):8458.

49. Alsuwian T, Usman MH, Amin AA. An Autonomous Vehicle Stability Control Using Active Fault-Tolerant Control Based on a Fuzzy Neural Network. Electronics (Basel). 2022 Oct 1;11(19):3165.

50. Qazi E ul H, Imran M, Haider N, Shoaib M, Razzak I. An intelligent and efficient network intrusion detection system using deep learning. Computers and Electrical Engineering. 2022 Apr;99:107764.

51. Qazi EUH, Almorjan A, Zia T. A One-Dimensional Convolutional Neural Network (1D-CNN) Based Deep Learning System for Network Intrusion Detection. Applied Sciences. 2022 Aug 10;12(16):7986.

52. Ahmad I, Ul Haq QE, Imran M, Alassafi MO, AlGhamdi RA. An Efficient Network Intrusion Detection and Classification System. Mathematics. 2022 Feb 8;10(3):530.

53. Bhati BS, Rai CS. Analysis of Support Vector Machine-based Intrusion Detection Techniques. Arab J Sci Eng. 2020 Apr 2;45(4):2371–83.

54. Ahmed N, Ngadi A bin, Sharif JM, Hussain S, Uddin M, Rathore MS, et al. Network Threat Detection Using Machine/Deep Learning in SDN-Based Platforms: A Comprehensive Analysis of State-of-the-Art Solutions, Discussion, Challenges, and Future Research Direction. Sensors. 2022 Oct 17;22(20):7896.

55. Tufan E, Tezcan C, Acarturk C. Anomaly-Based Intrusion Detection by Machine Learning: A Case Study on Probing Attacks to an Institutional Network. IEEE Access. 2021;9:50078–92.

56. Farrukh YA, Khan I, Wali S, Bierbrauer D, Pavlik JA, Bastian ND. Payload-Byte: A Tool for Extracting and Labeling Packet Capture Files of Modern Network Intrusion Detection Datasets. In: 2022 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT). IEEE; 2022. p. 58–67.

57. Ho CMK, Yow KC, Zhu Z, Aravamuthan S. Network Intrusion Detection via Flow-to-Image Conversion and Vision Transformer Classification. IEEE Access. 2022;10:97780–93.

58. Albasheer H, Md Siraj M, Mubarakali A, Elsier Tayfour O, Salih S, Hamdan M, et al. Cyber-Attack Prediction Based on Network Intrusion Detection Systems for Alert Correlation Techniques: A Survey. Sensors. 2022 Feb 15;22(4):1494.

59. Jiang Z, Zhang K, Xiang L, Yu G, Xu Y. A time-frequency spectral amplitude modulation method and its

applications in rolling bearing fault diagnosis. Mech Syst Signal Process. 2023 Feb;185:109832.

60. Xia X, Hashemi E, Xiong L, Khajepour A. Autonomous Vehicle Kinematics and Dynamics Synthesis for Sideslip Angle Estimation Based on Consensus Kalman Filter. IEEE Transactions on Control Systems Technology. 2023 Jan;31(1):179–92.

61. Rizvi S, Scanlon M, McGibney J, Sheppard J. Deep Learning Based Network Intrusion Detection System for Resource-Constrained Environments. In 2023. p. 355–67.

62. Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: ICISSP 2018 - Proceedings of the 4th International Conference on Information Systems Security and Privacy. SciTePress; 2018. p. 108–16.

63. IDS 2018 Intrusion CSVs (CSE-CIC-IDS2018) [Internet]. [cited 2024 May 10]. Available from: https://www.kaggle.com/datasets/solarmainframe/ids-intrusion-csv

64. Wang T, Lu C, Sun Y, Yang M, Liu C, Ou C. Automatic ECG Classification Using Continuous Wavelet Transform and Convolutional Neural Network. Entropy. 2021 Jan 18;23(1):119.

65. Djaballah S, Meftah K, Khelil K, Sayadi M. Deep Transfer Learning for Bearing Fault Diagnosis using CWT Time–Frequency Images and Convolutional Neural Networks. Journal of Failure Analysis and Prevention. 2023 Jun 21;23(3):1046–58.

66. Boateng EY, Otoo J, Abaye DA. Basic Tenets of Classification Algorithms K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review. Journal of Data Analysis and Information Processing. 2020;08(04):341–57.

67. Singh Kushwah J, Kumar A, Patel S, Soni R, Gawande A, Gupta S. Comparative study of regressor and classifier with decision tree using modern tools. Mater Today Proc. 2022;56:3571–6.

68. Hemeida AM, Hassan SA, Mohamed AAA, Alkhalaf S, Mahmoud MM, Senjyu T, et al. Nature-inspired algorithms for feed-forward neural network classifiers: A survey of one decade of research. Ain Shams Engineering Journal. 2020 Sep;11(3):659–75.

69. Laghrissi F, Douzi S, Douzi K, Hssina B. Intrusion detection systems using long short-term memory (LSTM). J Big Data. 2021 Dec 7;8(1):65.

70. Zulqarnain M, Ghazali R, Hassim YMM, Aamir M. An Enhanced Gated Recurrent Unit with Auto-Encoder for Solving Text Classification Problems. Arab J Sci Eng. 2021 Sep 22;46(9):8953–67.

71. MathWorks. Continuous 1-D wavelet transform [Internet]. [cited 2024 May 14]. Available from: https://www.mathworks.com/help/wavelet/ref/cwt.html

72. Alsemmeari RA, Dahab MY, Alsulami AA, Alturki B, Algarni S. Resilient Security Framework Using TNN and Blockchain for IoMT. Electronics (Basel). 2023 May 15;12(10):2252.

DATA AVAILABILITY STATEMENT

The dataset used in this research (CSE-CIC-IDS2018) is available in the following link: https://www.kaggle.com/datasets/solarmainframe/ids-intrusion-csv (accessed on 10 January 2024)

FUNDING

None.

CONFLICTS OF INTEREST

The authors declare no conflicts of interest.

AUTHORSHIP CONTRIBUTION

Conceptualization: Abdulaziz A. Alsulam.
Data curation: Abdulaziz A. Alsulam.
Formal analysis: Badraddin Alturki.

Research: Abdulaziz A. Alsulami and Badraddin Alturki.
Methodology: Abdulaziz A. Alsulami and Badraddin Alturki.
Project management: Abdulaziz A. Alsulami.
Resources: Badraddin Alturki.
Software: Abdulaziz A. Alsulami and Badraddin Alturki.
Supervision: Abdulaziz A. Alsulami.
Validation: Badraddin Alturki.
Display: Badraddin Alturki.
Drafting – original draft: Abdulaziz A. Alsulami and Badraddin Alturki.
Writing - proofreading and editing: Abdulaziz A. Alsulami and Badraddin Alturki.