SCT Proceedings in Interdisciplinary Insights and Innovations. 2025; 3:378
https://doi.org/10.56294/piii2025378

Category: Innovations in Science and Engineering

ORIGINAL

Enhancing cyber-attack prediction through optimized feature representation and advanced learning techniques

Mejora de la predicción de ciberataques mediante la representación optimizada de características y técnicas avanzadas de aprendizaje

Akkineni Yogitha[1] ✉, Bondili Sri Harsha Sai Singh[2] ✉.

[1] Department of Computer Science and Engineering, Velagapudi Ramakrishna Siddhartha Engineering College, Kanuru, Vijayawada, Andhra Pradesh, India.
[2] Department of Computer Science and Engineering, Koneru Lakshmaiah Educational Foundations, Vaddesswaram, Guntur, Andhra Pradesh, India.

Cite as: Akkineni Y, Bondili Sri Harsha SS. Enhancing cyber-attack prediction through optimized feature representation and advanced learning techniques. SCT Proceedings in Interdisciplinary Insights and Innovations. 2025;3:378. DOI: https://doi.org/10.56294/piii2025378

Submitted: 12-09-2025 **Reviewed:** 27-11-2024 **Accepted:** 03-01-2025 **Published:** 05-01-2025

Editor: Emanuel Maldonado

ABSTRACT

The integrity of computer networks and user security faces severe threats from web application attacks. Current threat detection techniques primarily rely on signature-based approaches, limiting their ability to recognize zero-day vulnerabilities. Moreover, the lack of comprehensive statistics on actual cyber-attacks further diminishes the effectiveness of these strategies. This paper introduces a comprehensive four-step methodology along with an architectural framework for the development of a robust cyberattack threat intelligence strategy. The initial phase involves data acquisition, encompassing the gathering of network traffic information and web page crawls, enabling the creation of feature vectors that effectively characterize cyber-attack information. Subsequently, the utilization of a sparse auto-encoder facilitates the analysis of the identified attack features. Finally, the proposed methodology incorporates the Convolutional Neural Network (ConvNNet) technique for systematic attack class prediction. Anomaly detection techniques are applied to forecast web-based attacks. The assessment leverages online cyber-attack datasets to evaluate the effectiveness of the proposed model. The original data yields a detection rate (DR) of 98.5% and a False Alarm Rate (FAR) of 9.5%. With training data, the model demonstrates an improved DR of 99% and a reduced FAR of 2%. Empirical analyses highlight the superior performance of the suggested approach compared to four competing machine learning methods, as evidenced by detection and false alarm rates across real-world and simulated web data.

Keywords: Fake News Detection, Natural Language Processing (NLP), Information Processing, Sentiment Analysis.

RESUMEN

La integridad de las redes informáticas y la seguridad de los usuarios se enfrentan a graves amenazas derivadas de los ataques a aplicaciones web. Las técnicas actuales de detección de amenazas se basan principalmente en enfoques basados en firmas, lo que limita su capacidad para reconocer vulnerabilidades de día cero. Además, la falta de estadísticas exhaustivas sobre ciberataques reales disminuye aún más la eficacia de estas estrategias. Este documento presenta una metodología integral de cuatro pasos junto con un marco arquitectónico para el desarrollo de una estrategia robusta de inteligencia sobre amenazas de ciberataques. La fase inicial consiste en la adquisición de datos, que abarca la recopilación de información sobre el tráfico de red y el rastreo de páginas web, lo que permite la creación de vectores de características que caracterizan eficazmente la información sobre ciberataques. Posteriormente, la utilización de un autocodificador disperso facilita el análisis de las características de ataque identificadas. Por último, la metodología propuesta incorpora la técnica de redes neuronales convolucionales (ConvNNet) para la predicción sistemática de clases de ataques. Se aplican técnicas de detección de anomalías para predecir los ataques basados en la web. La evaluación aprovecha conjuntos de datos de ciberataques en línea para evaluar la eficacia del modelo propuesto. Los datos originales arrojan una tasa de detección (DR) del 98,5% y una tasa de falsas alarmas (FAR) del 9,5%. Con los datos de entrenamiento, el modelo demuestra una RD mejorada del 99% y una FAR reducida del 2%. Los análisis empíricos destacan el rendimiento superior del enfoque propuesto en comparación con cuatro métodos de aprendizaje automático competidores, como demuestran las tasas de detección y falsas alarmas en datos web reales y simulados.

Palabras clave: Detección de noticias falsas, Procesamiento del Lenguaje Natural (PLN), Procesamiento de la Información, Análisis de Sentimiento.

INTRODUCTION

In recent decades, the rapid expansion of the internet has propelled it into a fundamental platform for global personal and professional operations. Given its pervasive influence, web services and programs have become indispensable for managing various physical and digital applications, consequently making them prime targets for exploitation by malicious entities, aiming to pilfer sensitive data and disrupt computing infrastructures [1]. Consequently, the incidence of cyberattacks on diverse online services and applications has significantly escalated [2].

According to recent data from Symantec, approximately 76% of all domains within the World Wide Web (WWW) exhibit vulnerabilities that require immediate patching [3]. Exploiting these vulnerabilities, malevolent actors frequently target user information, critical infrastructures, and even leverage gathered data for launching more sophisticated assaults.

These web-based attacks encompass a spectrum of tactics, including SQL injection [4], facilitating unauthorized data access, and cross-site scripting (XSS) attacks, compromising user sessions and embedding malware into vulnerable systems. To counteract these threats, various detection methods and countermeasures have been proposed and developed [5], [6].

Cyber attackers often exploit web services and applications to enhance social engineering techniques, such as phishing, leveraging users' inclination to establish trust online [7]. In these instances, attackers deceive victims into believing they are accessing legitimate websites, subsequently luring them into divulging sensitive information, such as passwords and banking details [8]. Protecting against these multifaceted cyber threats remains one of the foremost challenges for cybersecurity professionals.

The evolution of a defense-in-depth strategy has been necessitated by the sophisticated tactics employed by modern web hackers, advocating for the implementation of multiple security controls to effectively mitigate diverse attack techniques [9]. This approach incorporates layers of defense, including firewalls, intrusion detection systems, and access controls, to ensure comprehensive security for computer and network systems. It is imperative to establish autonomous defense mechanisms capable of adapting to the constantly evolving threat landscape in the event of cyberattacks [10].

Security techniques often involve a combination of signature-based, anomaly-based, and hybrid detection methods [11]. Signature-based methods monitor incoming traffic/activity patterns, comparing them against known attack signatures or patterns stored in a database or blacklist, triggering alerts upon detection of a matching pattern [12]. Contemporary security tools, such as firewalls, access controls, authentication, and intrusion monitoring systems, commonly employ these techniques. However, signature-based methods falter in identifying zero-day threats, i.e., attacks without predefined rules in the blacklist, requiring substantial computational resources for continuous rule and signature generation and updates. Anomaly-based detection methodologies, on the other hand, rely on establishing a comprehensive profile of the system's normal behaviour, thereby flagging any deviations from this profile as anomalous vectors [13]. By encompassing potential legitimate events within the profile, these techniques facilitate the detection of zero-day attacks at a lower computational cost compared to signature-based methods. Accordingly, this paper focuses on the development of an intelligent anomaly-based threat detection methodology for swift cyber-attack identification [14].

The effectiveness of existing web detection mechanisms hinges on the types of attacks being perpetrated. For instance, anti-phishing tools often utilize blacklists of known phishing websites as part of their signature-based approaches [15].

Malicious web applications for these blacklists are gathered from browsers and open sources such as PhishTank to train and test novel attack detection techniques. Anomaly-based detection plays a critical role in defending against various cyberattacks, including the detection of malicious SQL queries to prevent SQL injection attacks. Previous studies have addressed specific web application attacks, such as phishing, and summarized cyber-attack detection methodologies [17].

However, the need for robust web threat intelligence methodologies persists, necessitating the development of a comprehensive framework to mitigate these attacks and provide invaluable insights into their modus operandi.

To demonstrate the efficacy of modeling this framework and its potential applicability to real-world web applications, we propose a novel threat intelligence methodology for attacks based on statistical modeling of reliable web observations and data simulation. Our approach involves data gathering on actual cyberattacks and employs a sparse auto-encoder to dynamically extract crucial facets of cyberattacks. Subsequently, we advocate the use of a Convolutional Neural Network (ConvNNet) for the identification of current and imminent cyber-attacks, simulating attack data using the anomaly detection strategy and extracted features.

This research leverages the well-known Web Attack Dataset and UNSW-NB15 Dataset, representing valuable repositories of cyber-attack information. The following summarizes the primary contributions of this work:

- Introducing a streamlined approach for the automatic collection of web data and network traffic information, facilitating the extraction of useful features for the effective recognition of cyberattacks using a threat monitoring system.
- Exploring the impact of cyber data simulation on the efficacy of threat intelligence methodologies. Our simulation algorithm employs extracted features to model and accurately identify cyber-attacks.
- Proposing a novel Convolutional Neural Network (ConvNNet) strategy based on an anomaly detection methodology for the identification of web application attacks.

The remainder of this paper is organized as follows: Section 2 provides the context and reviews earlier research relevant to our work. Section 3 delineates the comprehensive methodology employed for data collection from cyberattacks and the extraction of valuable features for simulation purposes. Section 4 outlines the technique for modeling and simulating attack data using the extracted features. Finally, Section 5 concludes the work, offering insights into potential future directions for this field of study.

Present day AI models are growing into different new usecase, solving new Problems. So, current on Device DNN execution need to incorporate the additional model needs to be ready for realizing new usecase on Mobile devices. Eg. Regnet is a new model by facebook, we can check for implementation.
Arm NN is an inference engine for Arm CPUs, GPUs, and NPUs. Arm NN supports models created with TensorFlow Lite and ONNX frameworks.

Arm NN is the most performant machine learning (ML) inference engine for Android and Linux, accelerating ML on Arm Cortex-A CPUs and Arm Mali GPUs. Arm NN is written using portable C++14 and built using CMake - enabling builds for a wide variety of target platforms, from a wide variety of host environments. Python developers can interface with Arm NN through the use of our Arm NN TF Lite Delegate.

This ML inference engine is an open source SDK which bridges the gap between existing neural network frameworks and power-efficient Arm IP.

Open Neural Network Exchange (ONNX) is an open ecosystem that empowers AI developers to choose the right tools as their project evolves. ONNX provides an open source format for AI models, both deep learning and traditional ML. It defines an extensible computation graph model, as well as definitions of built-in operators and standard data types.

Currently we focus on the capabilities needed for inferencing (scoring). ONNX is widely supported and can be found in many frameworks, tools, and hardware. Enabling interoperability between different frameworks and streamlining the path from research to production helps increase the speed of innovation in the AI community.

LITERATURE REVIEW

The evolution of the smart grid in recent years has prompted the proposal of several Anomalous Detection Systems (ADS) for enhancing cyber-physical security, including multi-agent, model-based, machine learning-based, and protocol-specific ADS. However, these approaches only succeed in detecting concealed anomalies when equipped with multiple layers of measurement and management data.

Authors in [18] introduced a model-based ADS tailored for predicting data integrity attacks in automated generation control. Moreover, the use of redundant measurements to detect anomalies, as proposed in [19], is deemed inappropriate for Comprehensive Real-time Application Systems (CRAS), as unexpected disturbances may lead to increased estimation errors.

In the context of local defenses and Distributed Real-time Application Systems (DRAS), [20] demonstrated the effectiveness of a multi-agent-based ADS for identifying cyber-attacks. Several studies [21]-[23] have focused on synchrophasor communication protocols and an ADS for data acquisition and supervisory management based on specific requirements. While these signature-based ADSs excel at identifying anomalies in network packet logs, their rule creation requires a profound comprehension of communication protocols, rendering them less suitable for addressing big data challenges. Moreover, their inability to detect physical disturbances such as line faults limits their efficacy in ensuring the safety of CRAS.

The development of Wide-Area Monitoring Systems (WAMS) in ADS research, emphasizing the monitoring of frequency, voltage, and power oscillations while accommodating anomalies in incoming measurement signals, has been the subject of several investigations [24].

Pan et al. introduced an ADS based on common path mining to categorize cyber-attacks, regular activities, and physical disturbances, tested on a scaled-down two-line, three-bus transmission system. However, the use of "common path mining" entails manual labor and has limited applicability due to its slow detection rate. To accurately classify various events, numerous methods have been proposed for selecting relevant features, as the input characteristics of machine learning classifiers significantly influence their accuracy. The authors in [25] demonstrated the construction of Decision Tree (DT)-based classifiers for identifying system events using system parameters such as current, voltage, frequency, and derivative features. Utilizing the calculated derivative features of unprocessed Phasor Measurement Unit (PMU) data, different types of transient phenomena induced by power system cyber-physical events can be identified.

The application of Variational Mode Decomposition (VMD), an advanced signal decomposition technique, involves the separation of multi-component signals using various sub-signal patterns. Although it has shown superior performance compared to the traditional Empirical Mode Decomposition (EMD) method [26], its implementation lacks extensive discussion or insight into its potential to enhance CRAS cybersecurity. While it can facilitate multi-level classifications, this method may not be suitable for identifying other types of attacks, such as coordinated and communication-lacking assaults. In a groundbreaking study, authors in [27] provided novel insights into detecting false data injection attacks, establishing a Cyber-Physical Anomaly Detection System (CPADS) using the VMD method, encompassing a diverse range of occurrences, including natural disasters, cyber-attacks, and routine activities in the context of synchrophasor-based Wide-Area Phasor Applications (WAPS). This study's noteworthy feature is its exceptional capability to extract pertinent features, serving as the driving force behind its effectiveness. Notably, no prior research has been conducted on detecting coordinated cyber-attacks in CRAS cybersecurity [28].

Attackers are increasingly targeting the current state-of-the-art solution, as the control center-based protection function is susceptible to singular points of failure if compromised. Any form of abnormal malfunction poses a significant concern, as CRAS, designed to minimize power system disturbances, remains heavily reliant on exposed grid networks vulnerable to a multitude of cyber-attacks.

Previous studies have highlighted the vulnerability of critical applications such as WAPS to covert digital assaults [29]. In their research, the authors in [30] discussed the vulnerability assessment of SCADA communication protocols and the repercussions of deliberate internet-initiated attacks on RAS's transient voltage stability.

They provided examples of how covert and coordinated cyber-attacks, initiated by malware, can adversely affect the system's generator while attempting to impede RAS operations. The existing literature predominantly focuses on the analysis of measurement or management data surfaces, necessitating a thorough exploration of cyberattack surfaces at spatial and temporal layers of measurements and control signals to bolster CRAS's resilience against cyber threats.

Ultimately, this analysis will pave the way for the development of a defense-in-depth architecture. Previous investigations have scrutinized how covert intrusions impact control signals and have demonstrated the use of the decision tree (DT) for detecting malicious tripping attacks in CRAS.

DATA ENGINEERING

This section outlines the methodology employed for extracting features from cyber-attack data. Prior to delving into deep learning techniques for extracting relevant features from the collected data, we elucidate the process of acquiring web data through website crawling and network traffic capture.

Dataset

The process of gathering web data to facilitate feature extraction is illustrated in Figure 1. The figure delineates the diverse methods employed to acquire relevant information, encompassing data extracted from both network traffic and websites.

Leveraging web crawling technology, we collect HTML information and URLs from specific websites, prioritizing those structured in a manner that is user-friendly and easily comprehensible. Special attention is given to network attacks utilizing online services and phishing techniques.

For the Cyber Attack Dataset, we construct a whitelist of domains using web crawling, beginning with a random sample sourced from the DMOZ dataset. Additionally, 2,000 phishing URLs are selected randomly from a publicly accessible blacklist of phishing websites, with an indexed phishing URLs link stored in a CSV file.

As for the UNSW-NB15 Dataset, the feature vectors are stored in cloud technology. Our choice of this technology is attributed to its real-time database capability, high scalability, and support for a networked architecture suitable for both read- and write-intensive applications with SQL or NoSQL API access responsibilities.

Its capacity to handle big data in real-time extends to the management of disk-based and memory-optimized tables, the addition of servers to a live cluster, and the load balancing of data partitioning.

The efficacy of any cyber-attack modeling or detection system is contingent on the quality of the features characterizing the collected data.

An effective threat detection approach necessitates the dynamic generation of relevant features from the acquired data.

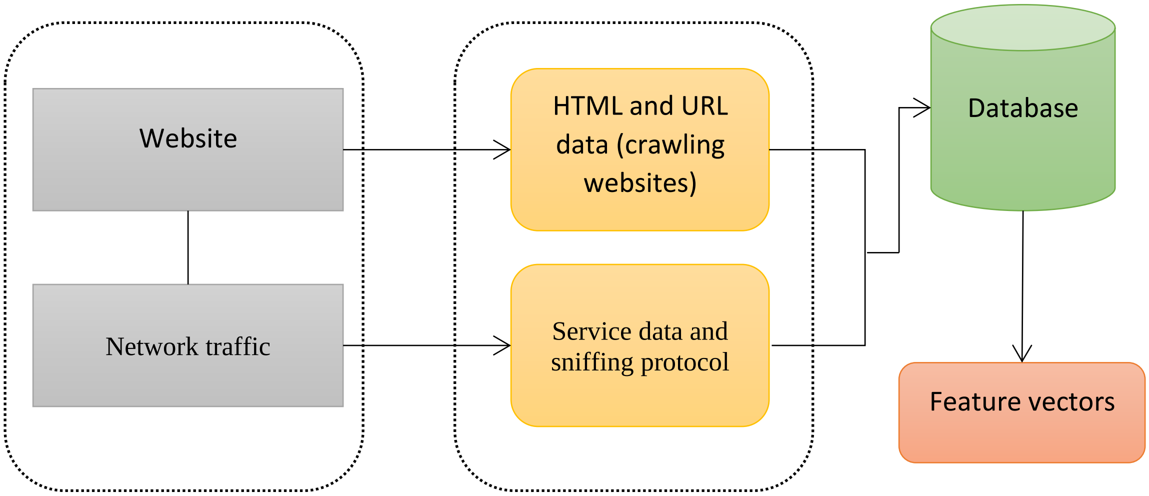
These features should comprehensively capture both the typical and anomalous behavior observed within the system. In our research, we employ the TCP/IP model to extract features, utilizing a comprehensive technique to investigate all services and application standards, such as (IP→TCP→HTTP).

Table 1 serves to identify aberrant web application behavior during our analysis. These features are derived from an in-depth examination of the application layer protocols and services within the TCP/IP model, notably focusing on prevalent methods for transmitting and receiving data between two locations, including DNS, HTTP, SMTP, FTP, and SNMP. The findings of our analysis underscore the significance of the outlined feature selection approach in enhancing the system's ability to identify web application attacks.

Table 1: Attributes related to Cyber-Attacks

Name	Explanation
url_words	Analysis of dynamic URL words and corresponding value mapping
Snmp_getrequest	Capturing the length of request messages using SNMP protocol
Snmp_duration	Recording the duration of SNMP protocol interactions
Length_smtp_subject	Content analysis of SNMP subject based on header
dns_qtype	Length analysis of query type using DNS protocol
dns_qclass	Length analysis of query class using DNS protocol
len_dnsanswer	Length of resource description in response to a DNS query
len_dnsquery	Length of DNS query
len_httphost	HTTP host header length analysis
len_httpurl	Length analysis of URL based on HTTP protocol

Fig 1 Feature extraction


Website
Network traffic
HTML and URL data (crawling websites)
Service data and sniffing protocol
Database
Feature vectors

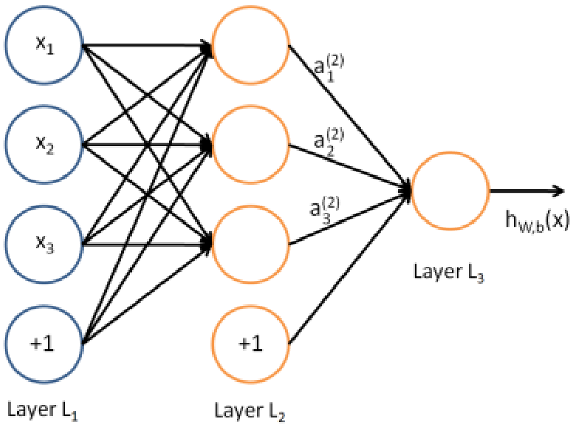
Feature Reduction

Unsupervised learning is achieved through the utilization of the Sparse Auto-encoder (SAE), commonly employed for dimensionality reduction and de-noising, offering robust nonlinear generalization. The SAE model is structured into three layers: an input layer, a hidden layer, and an output layer. Fig 2 illustrates the arrangement of these layers. The data undergoes reconfiguration, passing through the input layer, fewer hidden neurons, and ultimately returning to the input layer, facilitating feature extraction. The SAE learning process involves two primary steps: encoding and decoding. Encoding involves the nonlinear transformation of high-dimensional space into low-dimensional space. The input data X = {x_1, x_2,...,x_i} ∈ Z^n is transformed into a more abstract eigenvector H through the activation function f (•) of the encoder, as represented below:

$$H = f(W^T X + 'b)$$

Here, W denotes the weight matrix, b signifies the bias vector, and the function f (•) introduces non-linearity into the encoding process, ensuring the extraction of essential features within the data. Moreover, the decoder's activation function facilitates the reconstruction of the original data from the abstract representation H, seeking to minimize the reconstruction error. This reduction in dimensionality significantly aids in enhancing the efficiency and effectiveness of subsequent learning and prediction tasks, ultimately contributing to the overall robustness of the cyber-attack detection model.

Fig 2 Sparse auto-encoder



The decoder reconstructs the input from the eigenvector, reconstructing the input vectors Y = {y_1,y_2,...y_i} through the activation function f, the weight matrix W^((2)), and the bias term b^((2)), expressed as:

$$H = f(W^{(1)}X + b^{(1)}) \qquad (1)$$

Here, W denotes the weight matrix between different levels, while b represents the bias. Within the auto-encoder, the loss function J(W,b) during the decoding phase is computed as follows:

$$Y = f(W^{(2)}H + b^{(2)}) \qquad (2)$$

$$J(W,b) = \left(\frac{1}{M} \sum_{d=1}^{M} \frac{1}{2} \, ||x^d - \hat{x}^d \, ||\text{^}2 + \frac{\lambda}{2} \sum_{l=1}^{n_{l-1}} \sum_{i=1}^{S_l} \sum_{j=1}^{S_{l+1}} \left(W_{ji}^{(l)} \right)^2 \right) \qquad (3)$$

The Mean Squared Error (MSE) of the reconstruction constitutes the first component, while the regularization term aims to prevent overfitting. The weight decay coefficient λ, the total number of layers (NL), the total neurons (S_l), and the connection weight (W_ji^((l))) between neurons (i) in layer l + 1 and (j) in layer l are all indicated.

To minimize feature duplication and enhance the distinctiveness and identifiability of the encoded data, the Sparse Auto-encoder (SAE) incorporates the Kullback-Leibler (KL) divergence. The selective activation of neurons results in more distinct representations. The KL divergence is expressed as follows:

$$KL(\rho \| \hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j} \qquad (4)$$

$$\hat{\rho}_j = \frac{1}{M} \sum_{d=1}^{M} h_j^d \qquad (5)$$

Where \(\rho^_j = \frac{1}{M} \sum_{d=1}^{M} h_j^d \) represents the average activation level for each hidden unit. Here, M denotes the total number of hidden neurons, while ρ signifies the desired percentage of active neurons. Consequently, the loss function in the Sparse Auto-encoder (SAE) is formulated as:

$$J_{sparse}\,(W,b) = J(W,b) + \; \mu \sum_{j=1}^{M} KL\,(\rho \| \, \hat{\rho}_j) \qquad (6)$$

Here, μ represents the weight employed to regulate the penalty term (sparsity).

Learning Model

In light of the insights gleaned from the observations, the samples of the target classes "0" and "1" are balanced using the Synthetic Minority Over-sampling Technique (SMOTE).

The data trails' dimensionality is subsequently reduced through Principal Component Analysis (PCA).

The conclusive phase involves the utilization of a simplified Convolutional Neural Network (ConvNNet) architecture for predictive analysis.

Unlike the complex structures currently in use, the proposed architecture achieves the same level of accuracy at an accelerated rate.

Fig 3 Proposed ConvNNet architecture model

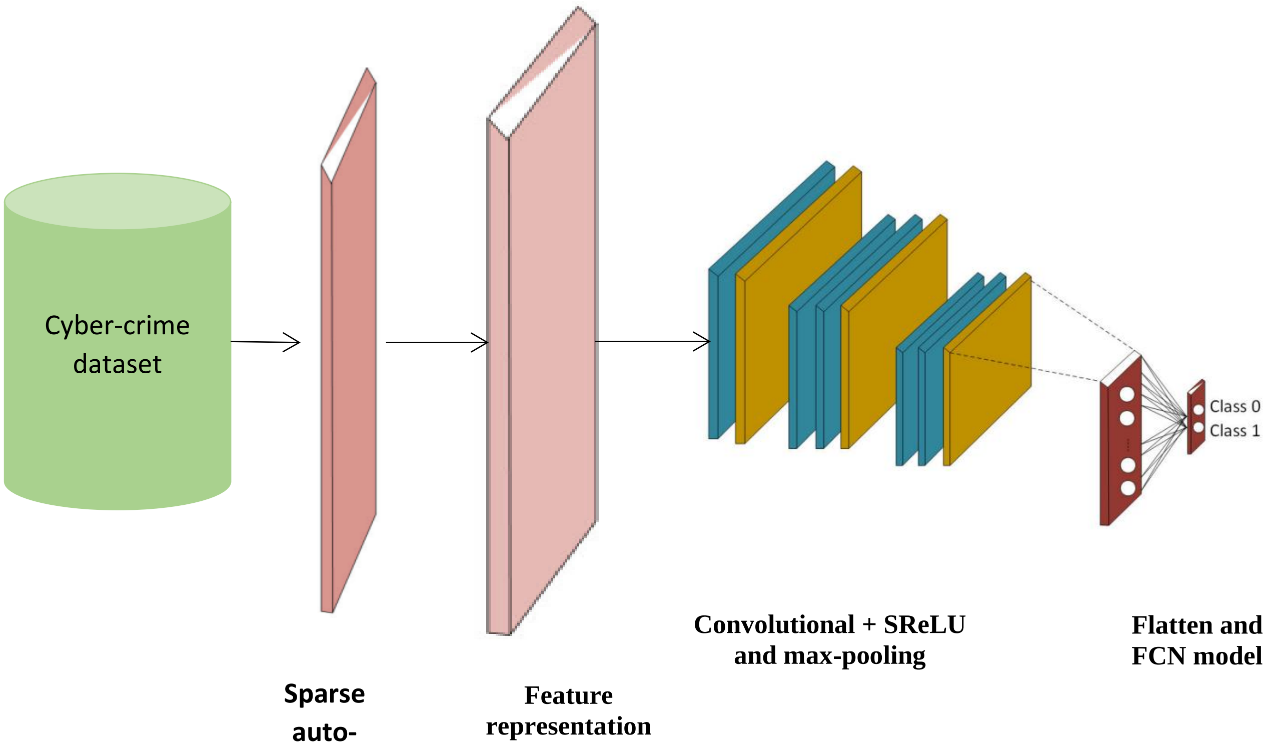


Fig 3 depicts the suggested straightforward ConvNNet design. According to Table 2, our suggested plan has three combinational layers, i.e., a pooling layer and two convolutional layers, each consisting of one combinational layer.

A convolutional layer is included in the final two combinational layers to extract more data before the pooling layer.

The suggested architecture consists of five convolutional layers with four, eight, sixteen, and sixteen filters, including kernel size and stride, each with a unique that allows the model to recognize the secret key bit.

There are four to eight different kernel sizes and strides. We have utilized max-pooling in our suggested design which chooses the most individuals from a given region. Its two main hyperparameters are speed and filter.

These hyperparameters remain unchanged throughout the learning process once they have been adjusted.

Table 2: ConvNNet Model

Layers	Output	No. of Parameters
conv1D_1	199*4	20
conv1D_2	66*8	135
conv1D_3	21*8	236
conv1D_4	6*16	530
conv1D_5	1*16	1040
conv1D_6	2	35

Table 3: Activation Functions Tested

Activation Function	Description
Scaled Residual Exponential	Demonstrates superior performance in feed-forward neural networks due to its self-normalizing capabilities.
Linear Unit (SReLU)	Network convergence is faster as internal normalization occurs more rapidly.
Rectified Linear Unit (ReLU)	Commonly used in deep learning models, enabling faster convergence during gradient descent.
Hyperbolic Tangent (tanh)	Functions effectively for classification tasks, providing a smooth gradient during training.

In the ConvNNet model, the value is set to 1-2. The convolutional layers employ various activation functions to address the non-linearity present in the data.

We experimented with different activation functions, and Table 3 outlines the different options tested during our analysis.

Notably, the Scaled Residual Exponential Linear Unit (SReLU) was chosen due to its superior performance in several classification tasks using feed-forward neural networks, attributed to its self-normalizing capabilities.

A key advantage of SReLU is the faster convergence of the network, facilitated by its swift internal normalization, which helps circumvent the issues of gradient vanishing and exploding.

Following the preceding max-pooling layer, the resultant column vector is connected to the fully connected layer. Backpropagation is then employed during each training iteration within an epoch. The final layer of the fully connected layer outputs a vector with N dimensions, where N represents the total output target classes, in our case, N = 2. The output from the preceding uniform layer serves as the input. After several epochs of training, the models are capable of classifying the provided features utilizing the softmax technique. After training the model for 200 epochs over an extended period, we have sufficient batch training cycles to evaluate its performance. Our findings indicate that the model's performance stabilizes before reaching 50 epochs.

To expedite learning and facilitate the convergence of the model, we applied standardization to mitigate the effects of wide variations between the data's minimum and maximum values. This process aids in the rapid convergence and accurate development of the model.

Overfitting, which may occur due to noisy side-channel leaks, was addressed using specific strategies such as regularization and dropout techniques. Through testing, we determined that L2 regularization produced superior outcomes, effectively controlling and minimizing the weights to prevent overfitting.

Additionally, we eliminated duplicate occurrences in the training dataset to prevent overfitting and biased model development resulting from learning from noise. The hyperparameters associated with each layer were adjusted to enhance the ConvNNet's efficiency, with the optimal model parameters selected through an exhaustive search process utilizing the grid search feature within the Scikit-learn library.

In order to examine and model attack behaviors, cyber simulation has become increasingly vital, allowing for the replication and emulation of cyber-attack scenarios. Simulation results can aid in system vulnerability assessments and provide insight into the potential impacts of such attacks. In our research, we opted to use dynamic features for simulating actual cyber-attack data, enabling us to test the efficacy of web detection methods without engaging in actual cyber data assaults.

Data simulation involves generating numerous random examples that adhere to a specific distribution, with the mean and standard deviation estimated for both genuine and deceptive data. To generate new values within the original real data range, we define the feature range for web data. Additional data points are created using the Probability Density Function (PDF) equation for the normal distribution based on the supplied X sequence numbers and the necessary number of vectors.

A statistical analysis of the real-world and synthetic cases is provided through a plot to determine if any variations exist between the instances. This plot serves to ascertain whether a data set adheres to a statistically normal distribution, as illustrated in Fig []. The close resemblance between the theoretical distribution lines of the two cases indicates that simulated web data can indeed enhance threat detection methods.

The ConvNNet technique was implemented using the MATLAB 2020a programming language on a computer equipped with an i7 processor and 16 GB RAM. Two actual attack datasets were employed to evaluate the proposed approach. One of these datasets was obtained from a subset of blacklisted website information sourced from PhishTank, a community-driven website that maintains blacklists of phishing URLs based on user submissions and verifications. Another dataset was gathered from the UNSW-NB15 dataset, containing various recent malicious and legitimate network instances. The evaluation of the model's performance and comparison with competing methods was conducted using this dataset. We refer to the dataset collected from PhishTank as the "Cyber attack dataset" throughout the remaining sections of this article.

For the training segment, the selection of normal samples accounts for approximately 57% to 70% of the total samples, with the remaining samples used for testing. Tenfold cross-validation was implemented to assess the method's effectiveness and prevent bias favoring the majority class. The MATLAB 2020a library toolkit was employed to develop the ConvNNet method, dynamically selecting the most crucial elements of web information. The technique's parameters, represented as mins up and min conf, were tuned using three values (0.4, 0.6, and 0.8) to select the most robust rules. The top features were then selected based on these guidelines.

The results were classified into three categories: low, median, and high. Several principles and their relevance scores for both datasets were illustrated in Fig 7, calculated as the average of each rule's support and confidence.

The dynamic feature extraction from the URLs in the cyber-attack dataset was depicted in Fig 4 to Fig 11, where the arranged words of each URL observation were denoted as (W_1, W_2,...W_n). Additionally, the rule set in Fig 5 incorporated 47 characteristics from the UNSW-NB15 dataset, selecting features of utmost importance based on Table 4.

Table 4 presents the overall performance evaluation of various approaches, including CART, k-NN, SVM, RF, GM, and ConvNNet, on both the Web attack dataset and the UNSW-NB15 dataset in their original form. The evaluation metrics include Detection Rate (DR) and False Alarm Rate (FAR) expressed as percentages for each approach.

Approaches	Web attack dataset (Original dataset)	UNSW-NB15 dataset (Original dataset)
CART	94.5	90
k-NN	92.5	88
SVM	92.6	91
RF	95.4	93
GM	97	96
ConvNNet	98.5	98

As depicted in the table, the ConvNNet approach demonstrates superior performance in terms of the Detection Rate (DR) on both datasets, achieving 98.5% on the Web attack dataset and 98% on the UNSW-NB15 dataset.

However, there is a slightly higher False Alarm Rate (FAR) associated with the ConvNNet approach, indicating a higher rate of false positives, particularly on the UNSW-NB15 dataset.

Comparatively, the Generalized Model (GM) also shows competitive performance, achieving 97% DR on the Web attack dataset and 96% DR on the UNSW-NB15 dataset, with relatively lower false positive rates compared to ConvNNet. Random Forest (RF) exhibits strong performance as well, achieving a high DR with a relatively low FAR on both datasets. Support Vector Machine (SVM) and Classification and Regression Trees (CART) demonstrate moderate performance, with a DR ranging between 90-94.5% and a FAR between 7-10%. The k-Nearest Neighbors (k-NN) approach displays the lowest performance among the methods evaluated, with a DR between 88-92.5% and a FAR between 8-11.8%.

Overall, the results suggest that the ConvNNet model shows the most robust performance, particularly in terms of the Detection Rate, making it a promising approach for cyber-attack detection in both the Web attack dataset and the UNSW-NB15 dataset. However, further optimization may be necessary to reduce the False Alarm Rate for improved accuracy.

The regulations, selected based on their priority according to the natural operation of the model, represent the most significant features. For this assessment, the significance threshold was maintained at 0.6 (high), and any attribute with an importance score exceeding 0.7 for each dataset was selected using the ConvNNet method.

Certain features from both datasets are illustrated in **Fig 4 to Fig 11**, where the extracted features are made understandable to users through tokenization during the pre-processing step, such as the use of hostnames like "Google." These comprehensible features could potentially be utilized in the future to enhance phishing awareness and detection mechanisms. Fig 8 illustrates the features selected from the UNSW-NB15 dataset, where the captured network traffic exhibited services and protocols corresponding to these specifications.

To provide the reader with a clearer understanding of how the ConvNNet method performed in comparison to four competing methods, the overall DR and FAR for each method were calculated using the original data. Table 5 presents the results for CART, k-NN, SVM, and RF across both datasets.

FINDINGS AND RESULTS

Evaluating the performance of a network intrusion detection system using the UNSW-NB15 dataset typically involves a range of evaluation metrics. Here's a list of commonly used evaluation metrics, along with descriptions, and I'll also provide information on how to create appropriate graphs for some of them:

True Positive (TP): The number of actual attacks correctly identified by the system.

True Negative (TN): The number of normal instances correctly identified as normal by the system.

False Positive (FP): The number of normal instances incorrectly identified as attacks by the system (Type I Error).

False Negative (FN): The number of actual attacks that the system failed to identify (Type II Error).

Accuracy: The proportion of correct classifications, calculated as (TP + TN) / (TP + TN + FP + FN).

Precision (Positive Predictive Value): The ability of the system to correctly identify attacks when it claims they are attacks, calculated as TP / (TP + FP).

Recall (Sensitivity, True Positive Rate): The ability of the system to correctly identify attacks out of all actual attacks, calculated as TP / (TP + FN).

Specificity (True Negative Rate): The ability of the system to correctly identify normal instances out of all actual normal instances, calculated as TN / (TN + FP).

F1-Score: The harmonic mean of precision and recall, which balances precision and recall. It's calculated as 2 * (Precision * Recall) / (Precision + Recall).

Area Under the ROC Curve (AUC-ROC): This metric measures the ability of the model to distinguish between attack and normal instances. You can plot a Receiver Operating Characteristic (ROC) curve to visualize this and calculate the AUC.

Area Under the Precision-Recall Curve (AUC-PR): This metric assesses the trade-off between precision and recall and is particularly useful when dealing with imbalanced datasets. You can plot a Precision-Recall curve to visualize this and calculate the AUC.

The graphs below demonstrate the relationship between FAR and DR, providing a clear visualization of how these techniques can be applied in practice.

Fig 4 DR evaluation with web attack dataset based on original data

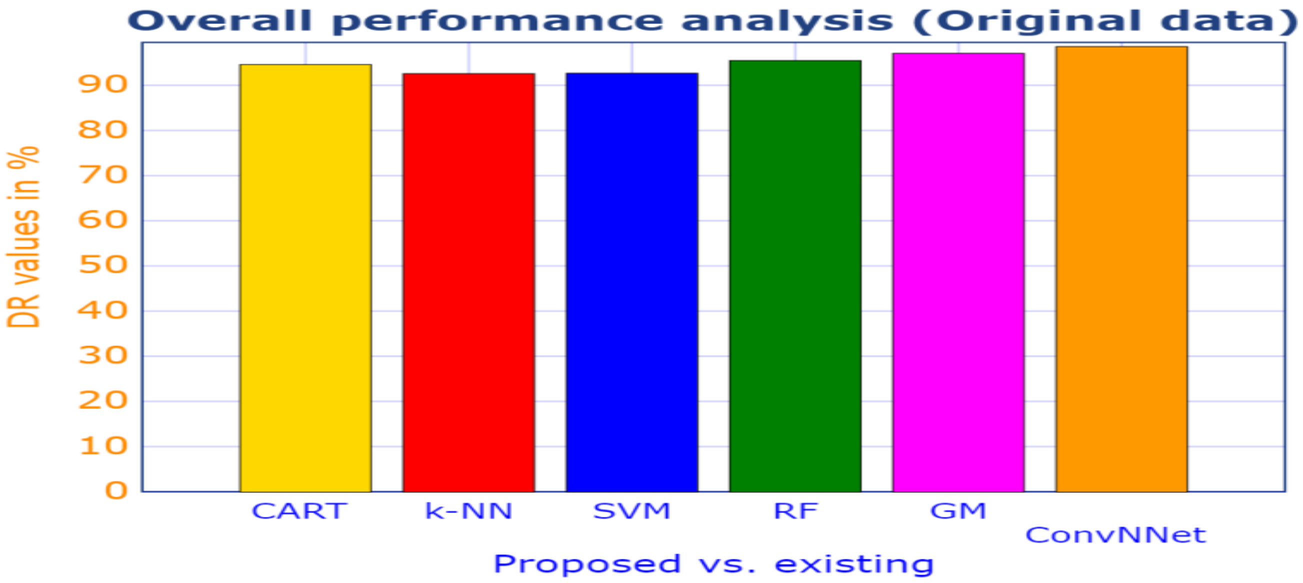
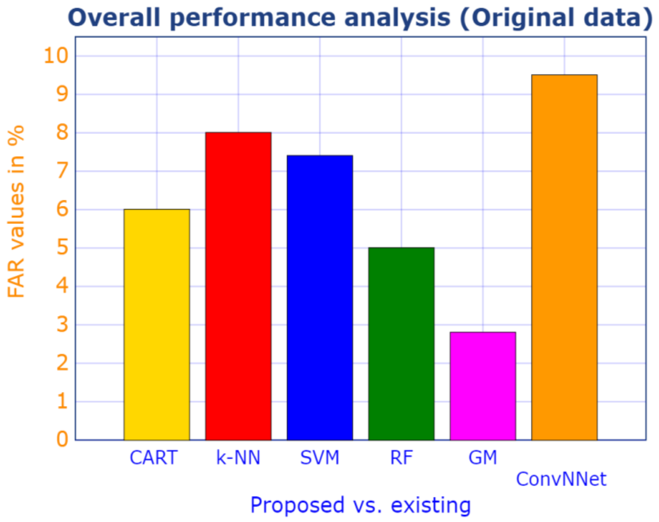


Fig 5 DR evaluation with UNSW-NB15 dataset based on original data



https://doi.org/10.56294/piii2025378

Fig 6 FAR evaluation with web attack dataset based on original data

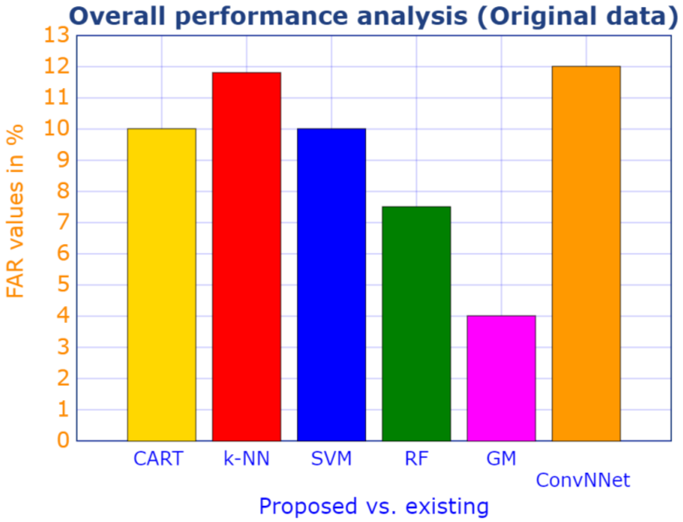


Fig 7 FAR evaluation with UNSW-NB15 dataset based on original data

These results emphasize the potential workflow and practical implications of utilizing these techniques in the context of cyber-attack detection.

They showcase the trade-offs between detection rates and false alarm rates, providing valuable insights into the overall performance of each method and their suitability for real-world application.

Comparing the performance of the ConvNNet method with the competing approaches on both datasets reveals its superiority. In the case of the cyber-attack dataset, the ConvNNet method outperforms the other techniques with a False Alarm Rate (FAR) ranging between 5% to 8% and an average accuracy of 95%.

Similarly, when considering the initial data from the UNSW-NB15 dataset, the ConvNNet method exhibits a higher Detection Rate (DR) of 95.56% and a lower FAR of 4.5%, surpassing the competitors' average FAR of 4% to 12% and DR of 88% to 92%, respectively.

The feature extraction process for the cyber-attack dataset involves meticulous analysis of various features, while the second feature selection from the UNSW-NB15 dataset is automatically derived from the URLs, distinguishing between normal and attack observations based on protocols and services.

Conducting a statistical analysis on the UNSW-NB15 dataset to contextualize the feature quality using the ConvNNet technique reveals that the average DR is notably lower at approximately 75% compared to others, as illustrated in Table 2, where the ConvNNet method demonstrates an average DR of around 91% despite utilizing 49 features.

The ConvNNet method generates high-quality features that enhance the capability to detect web application attacks, as evidenced by the comparison between the two datasets.

These findings underscore the ConvNNet method's efficacy and its potential to improve cyber-attack detection in practical settings.

Approaches	Web attack dataset (Original dataset)	UNSW-NB15 dataset (Original dataset)
CART	94.5	90
k-NN	92.5	88
SVM	92.6	91
RF	95.4	93

Approaches	Web attack dataset (Original dataset)	UNSW-NB15 dataset (Original dataset)
GM	97	96
ConvNNet	98.5	98

Fig 8 DR evaluation with web attack dataset based on training data

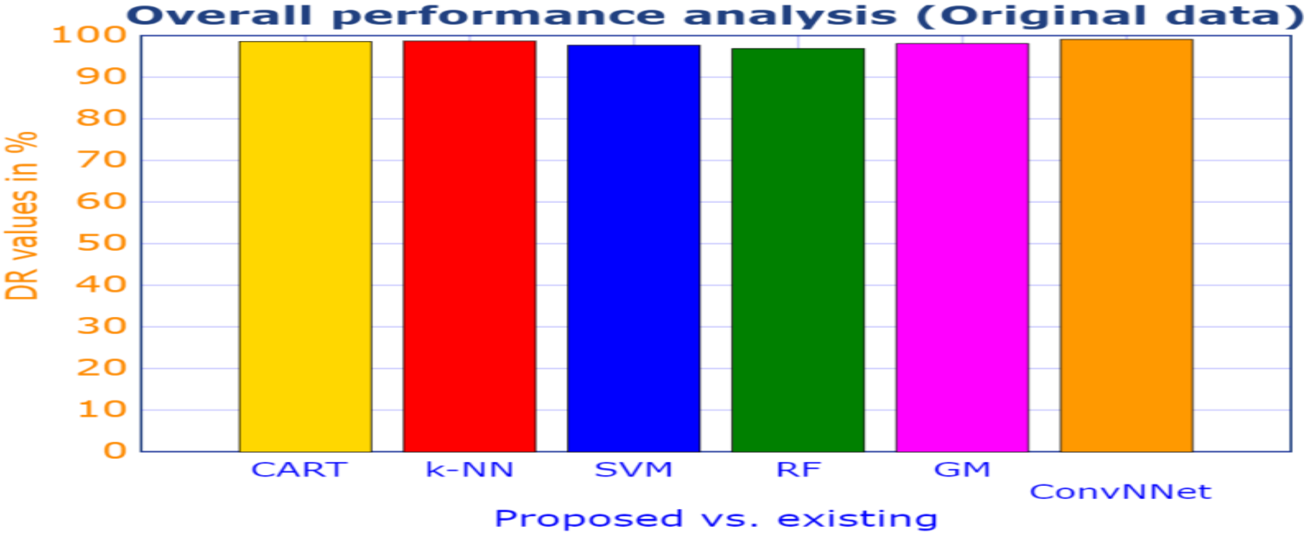
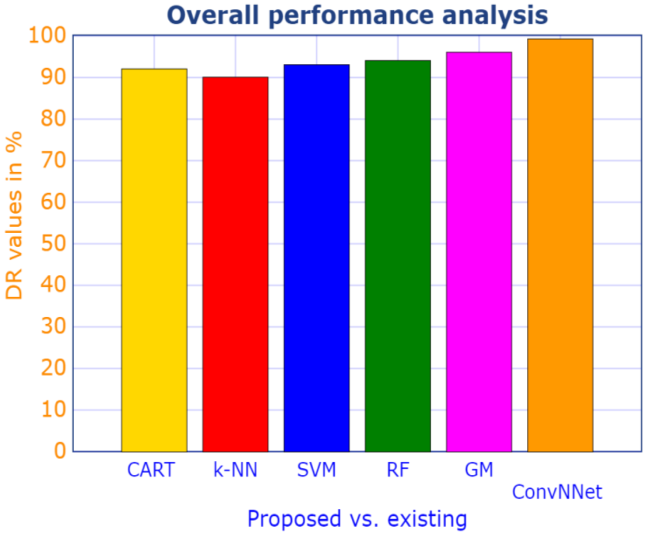


Fig 9 DR evaluation with UNSW-NB15 based on training data



https://doi.org/10.56294/piii2025378

Fig 10 DR evaluation with web attack dataset based on training data

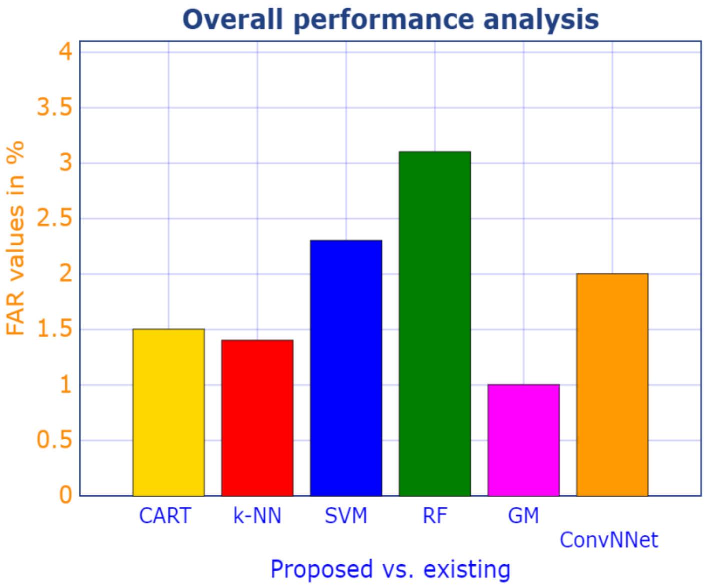
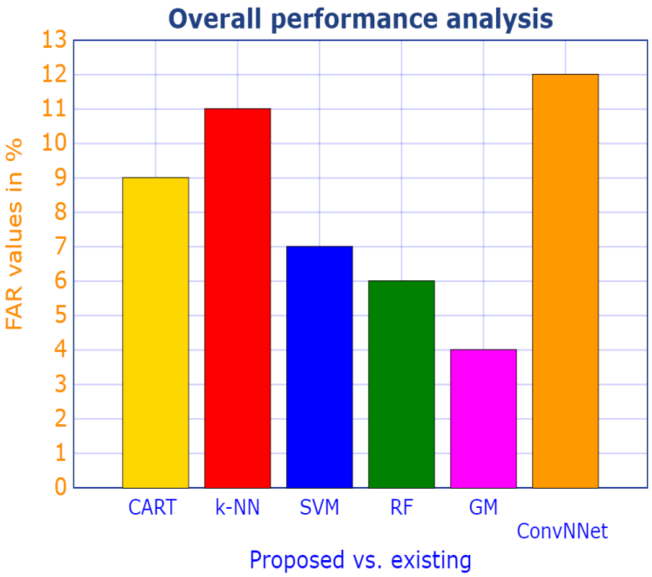


Fig 11 DR evaluation with UNSW-NB15 based on training data



Based on the comparison presented in Table 2, it is evident that the ConvNNet approach performs significantly better than the other four methods on the simulated data. The charts representing the Detection Rate (DR) and False Alarm Rate (FAR) are displayed in Figures 8 to 11. Specifically, when applied to the original data from the cyber-attack dataset, the ConvNNet method achieves a DR of 98.68% and a FAR of 1.04%, surpassing the average accuracies of the other methods, which range from 96% to 98%, with FARs between 1.4% and 3.1%.

Similarly, when utilizing the UNSW-NB15 dataset, the ConvNNet method outperforms its competitors, yielding a DR of 95.68% and a FAR of 4.32%, compared to the other methods' average DR of 90% and FAR of 10%.

Furthermore, it is worth noting that the synthetic data, which is generated based on the specifications of each feature's lower and upper bounds and drawn from a Gaussian distribution, consistently achieves a 1-2% higher DR compared to the original data on both datasets.

This increase in performance can be attributed to the careful control over the variability of the numbers in the synthetic dataset.

The discussion provides a comprehensive overview of the findings from the evaluation of the ConvNNet method, highlighting its ability to accurately identify pertinent attributes from network and web application data.

This capability is crucial in enhancing cyber attack detection mechanisms, as different types of cyber attacks present distinct challenges, and the ConvNNet method can effectively address these challenges.

Furthermore, the discussion emphasizes the importance of utilizing the ConvNNet method for automated feature extraction from websites, thereby enhancing the scalability and efficiency of cyber attack detection tools. By simulating data from both the cyber attack dataset and the UNSW-NB15 dataset, the ConvNNet method demonstrates its capability to accurately replicate both legitimate and malicious web data, thereby providing a robust solution for improving cyber attack threat intelligence.

The discussion also acknowledges that while the results from the cyber attack dataset were slightly superior to those from the UNSW-NB15 dataset, both datasets effectively contribute to identifying web application attacks using the ConvNNet technique. The method's ability to select feature values from the dataset, whether real or synthetic, underscores its potential in safeguarding current web applications and their services.

Moreover, the discussion highlights the superiority of the ConvNNet approach over other methods, particularly in terms of DR, FAR, and processing time. The method's reliance on statistical design and thresholds for normal data ensures its effectiveness in real-time cyber attack detection scenarios, even without prior knowledge of specific attack vectors.

However, the discussion also acknowledges certain limitations, such as the need for conversion of nominal characteristics into numerical features and the method's inability to define specific attack types. Addressing these limitations and conducting further research will be essential to adapt the ConvNNet method for practical applications.

CONCLUSIONS

The conclusion summarizes the key findings and contributions of the research. It highlights the successful development and evaluation of a novel ConvNNet method for simulating and identifying various cyberattacks, both known and unknown.

The method is built on dynamic website analysis and network data, enabling the automatic extraction of essential features for cyber attack detection.

The evaluation of the proposed approach using actual and simulated data from the UNSW-NB15 and cyber attack datasets demonstrates its superior performance in comparison to four other machine-learning techniques.

The simulated data highlights the potential of using MATLAB 2020a to generate attack data for web applications, revealing an increase in detection rates and false alarm rates.

The research findings indicate that the proposed approach can effectively address the information gap associated with cyberattacks, making it challenging to develop and evaluate detection mechanisms.

The high detection rates and low false alarm rates achieved by the ConvNNet method, both with original and training data, demonstrate its potential for enhancing cyber attack detection and prevention strategies.

The conclusion also points to future research directions, including the application of the developed scheme in real social network settings and the testing of advanced cyberattack types. Additionally, the exploration of more specialized feature sets to enable a detailed analysis of specific attack types is suggested as a potential avenue for further investigation.

REFERENCES

[1] T. Gao, F. Li, Y. Chen, and X. Zou, "Preserving local differential privacy in online social networks," in International Conference on Wireless Algorithms, Systems, and Applications. Springer, 2017, pp. 393–405.

[2] R. Girshick, "Faster R-CNN: Towards real-time object detection with region proposal networks," in Advances in Neural Information Processing Systems, 2015, pp. 91–99.

[3] Z. Levi and T. Hassner, "Age and gender classification using convolutional neural networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2015, pp. 34–42.

[4] M. Khadangi and M. H. F. Zarandi, "From type-2 fuzzy rate-based neural networks to social networks' behaviors," in 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), July 2016, pp. 1970–1975.

[5] Z. Li, D. y. Sun, J. Li, and Z. f. Li, "Social network change detection using a genetic algorithm based backpropagation neural network model," in 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Aug 2016, pp. 1386-1387.

[6] A. Luna, M. N. del Prado, A. Talavera, and E. S. Holguín, "Power demand forecasting through social network activity and artificial neural networks," in 2016 IEEE ANDESCON, Oct 2016, pp. 1–4.

[7] I. Habernal, T. Pta'cek, and J. Steinberger, "Sentiment analysis in Czech social media using supervised machine learning," in Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis, 2013, pp. 65–74.

[8] M. M. Hasan, M. M. Islam, M. I. I. Zarif, and M. M. A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches," Internet Things, vol. 7, Sep. 2019, Art. no. 100059.

[9] Sriram, "An efficient intrusion detection system based on hypergraph Genetic algorithm for parameter optimization and feature selection in support vector machine," Knowledge-Based Systems, vol. 134, pp. 1-12, Oct. 2017.

[10] Bhatt, E. T. Yano, and P. Gustavsson, "Towards a framework to detect multi-stage advanced persistent threats attacks," in Proc. IEEE 8th Int. Symp. Service Oriented Syst. Eng., Apr. 2014, pp. 390–395.

[11] Mazraeh, M. Ghanavati, and S. H. N. Neysi, "Intrusion detection system with decision tree and combine method algorithm," International Academic Journal of Science and Engineering, vol. 3, no. 8, pp. 21–31, 2016.

[12] Guo, Y. Ping, N. Liu, and S.-S. Luo, "A two-level hybrid approach for intrusion detection," Neurocomputing, vol. 214, pp. 391–400, Nov. 2016.

[13] Friedberg, F. Skopik, G. Settanni, and R. Fiedler, "Combating advanced persistent threats: From network event correlation to incident detection," Computers & Security, vol. 48, pp. 35–57, Feb. 2015.

[14] Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in Proc. International Conference on Signal Processing, Communication, and Engineering Systems, Jan. 2015, pp. 92–96.

[15] Johnson and E. A. Hogan, "An analytic graph metric for mitigating advanced persistent threat," in Proc. IEEE International Conference on Intelligent Security Information, Jun. 2013, pp. 129–133.

[16] Aziz, A. E. Hassanien, S. E.-O. Hanaf, and M. F. Tolba, "Multilayer hybrid machine learning techniques for anomalies detection and classification approach," in Proc. 13th International Conference on Hybrid Intelligent Systems (HIS), Dec. 2013, pp. 215–220.

[17] Aziz, H. A. El-Mousad, S. E.-O. Hanaf, and M. F. Tolba, "Multilayer hybrid machine learning techniques for anomalies detection and classification approach," in Proceedings of the 13th International Conference on Hybrid Intelligent Systems (HIS), December 2013, pp. 215–220.

[18] Wang, Y. Li, Z. Chen, P. Zhang, and G. Zhang, "A survey of exploitation techniques and defenses for program data attacks," Journal of Network and Computer Applications, vol. 154, March 2020, Article no. 102534.

[19] Singh, P. K. Sharma, S. Y. Moon, D. Moon, and J. H. Park, "A comprehensive study on APT attacks and countermeasures for future networks and communications: Challenges and solutions," Journal of Supercomputing, vol. 75, no. 8, pp. 4543–4574, August 2019.

[20] Alshamrani, S. Myneni, A. Chowdhary, and D. Huang, "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities," IEEE Communications Surveys & Tutorials, vol. 21, no. 2, pp. 1851–1877, 2nd Quarter, 2019.

[21] Sherubha, "Graph-Based Event Measurement for Analyzing Distributed Anomalies in Sensor Networks," Sådhanå (Springer), 45:212, doi: 10.1007/s12046-020-01451-w.

[22] Sherubha, "An Efficient Network Threat Detection and Classification Method using ANP-MVPS Algorithm in Wireless Sensor Networks," International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-8 Issue-11, September 2019.

[23] Sherubha, "An Efficient Intrusion Detection and Authentication Mechanism for Detecting Clone Attack in Wireless Sensor Networks," Journal of Advanced Research in Dynamical and Control Systems (JARDCS), Volume 11, issue 5, Pg No. 55-68.

[24] Cai, N. Meng, B. Ryder, and D. Yao, "DroidCat: Effective Android malware detection and categorization via app-level profiling," IEEE Transactions on Information Forensics and Security, vol. 14, no. 6, pp. 1455–1470, June 2019.

[25] Moustafa, J. Hu, and J. Slay, "A holistic review of network anomaly detection systems: A comprehensive survey," Journal of Network and Computer Applications, vol. 128, pp. 33–55, February 2019.

[26] Peng, S. Yu, and A. Yang, "Smartphone malware and its propagation modeling: A survey," IEEE Communications Surveys & Tutorials, vol. 16, no. 2, pp. 925–941, 2nd Quarter, 2014.

[27] Delac, M. Silic, and J. Krolo, "Emerging security threats for mobile platforms," in Proceedings of the MIPRO 34th International Convention on Information and Communication Technology, Electronics, and Microelectronics, May 2011, pp. 1468–1473.

[28] Yang, J. Zhang, and G. Gu, "Understanding the market-level and network-level Behaviors of the Android malware ecosystem," in Proceedings of the IEEE 37th International Conference on Distributed Computing Systems (ICDCS), June 2017, pp. 2452–2457.

[29] S.-H. Seo, D.-G. Lee, and K. Yim, "Analysis on maliciousness for mobile applications," in Proceedings of the 6th International Conference on Innovative Mobile Internet Services in Ubiquitous Computing, July 2012, pp. 126–129.

[30] Gao and J. Liu, "Modeling and restraining mobile virus propagation," IEEE Transactions on Mobile Computing, vol. 12, no. 3, pp. 529–541, March 2013.

FINANCING

None.

CONFLICT OF INTEREST

None.

https://doi.org/10.56294/piii2025378