Journal of Cloud Computing

RESEARCH Open Access

Intrusion detection in cloud computing based on time series anomalies utilizing machine learning

Abdel-Rahman Al-Ghuwairi[1], Yousef Sharrab[2], Dimah Al-Fraihat[3*], Majed AlElaimat[1], Ayoub Alsarhan[4] and Abdulmohsen Algarni[5*]

Abstract
The growth of cloud computing is hindered by concerns about privacy and security. Despite the widespread use of network intrusion detection systems (NIDS), the issue of false positives remains prevalent. Furthermore, few studies have approached the intrusion detection problem as a time series issue, requiring time series modeling. In this study, we propose a novel technique for the early detection of intrusions in cloud computing using time series data. Our approach involves a method for Feature Selection (FS) and a prediction model based on the Facebook Prophet model to assess its efficiency. The FS method we propose is a collaborative feature selection model that integrates time series analysis techniques with anomaly detection, stationary, and causality tests. This approach specifically addresses the challenge of misleading connections between time series anomalies and attacks. Our results demonstrate a significant reduction in predictors employed in our prediction model, from 70 to 10 predictors, while improving performance metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), Median Absolute Percentage Error (MdAPE), and Dynamic Time Warping (DTW). Furthermore, our approach has resulted in reduced training, prediction, and cross-validation times of approximately 85%, 15%, and 97%, respectively. Although memory consumption remains similar, the utilization time has been significantly reduced, resulting in substantial resource usage reduction. Overall, our study presents a comprehensive methodology for effective early detection of intrusions in cloud computing based on time series anomalies, employing a collaborative feature selection model and the Facebook Prophet prediction model. Our findings highlight the efficiency and performance improvements achieved through our approach, contributing to the advancement of intrusion detection techniques in the context of cloud computing security.

Keywords Intrusion Detection, Cloud Computing Security, Features Selection, Time Series Analysis, Granger Causality, Facebook Prophet, Machine Learning

*Correspondence:
Dimah Al-Fraihat
d.fraihat@iu.edu.jo
Abdulmohsen Algarni
a.algarni@kku.edu.sa
[1] Department of Software Engineering, The Hashemite University, Zarqa, Jordan
[2] Department of Data Science and Artificial Intelligence, Isra University, Amman, Jordan
[3] Department of Software Engineering, Isra University, Amman, Jordan
[4] Department of Information Technology, The Hashemite University, Zarqa, Jordan
[5] Department of Computer Science, King Khalid University, Abha, Saudi Arabia

© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Introduction

Cloud computing, defined as internet-based computing in which virtual shared servers and workstations supply software, infrastructure, platforms, and resources, is a constantly growing field [1, 2]. Cloud computing is vulnerable to attacks due to its open structure. As a result, privacy and security are critical to the success of cloud computing [3]. Conventional security procedures are incapable of offering appropriate solutions to this challenge. For example, the Denial of Service (DoS) and the Sybil attack are unavoidable assaults [4]. One of the oldest and most critical cyber security concerns is the identification of hostile network activities. Because of the variety of technologies used and the expansion of networks, issues related to security become more complex. Additionally, odd network behaviour is misinterpreted such as signalling a hacked device or network [5].

For protecting cloud environments against various threats and cyberattacks, intrusion detection systems (IDS) have become the most extensively used component of computer systems security and security procedures [6]. IDS use a range of reaction mechanisms to discover vulnerabilities, identify illegal activities, and perform prevention measures in order to stay up with the progress of computer-related crimes [7]. However, a recent review research reported that little research has portrayed the intrusion detection problem as a time series problem [8].

Features Selection (FS) is an essential preprocessing phase that has become popular in network administration, especially for detecting network intrusions and traffic classification issues. In cybersecurity, FS is often used with algorithms of Machine Learning (ML) to build predictive models for intrusion detection, network anomaly detection, and malware categorization, among other security requirements [7]. The idea is to identify the most relevant features from large and complex datasets that capture the behaviour and patterns of cyber-attacks. The use of FS can enhance the performance of ML algorithms by reducing the dimensionality of the dataset, increasing the training speed, and improving the generalization of new data. However, existing ML detection algorithms depend mainly on basic association elements to establish the link between attacks and traffic parameters. Causal reasoning demonstrates that failing to distinguish between correlation and causation may lead to diagnostic mistakes [8].

Existing research utilizes intrusion detection and feature selection of frequently obsolete data that does not account for recent cyber-attacks [8]. The majority of researchers used the “KDDCUP99” and “NSL-KDD” datasets for their Features Selection, which produced significant results concerning accuracy and detection rate. However, to produce a thorough comparison in the detection of network attacks, a comprehensive and representative dataset must be used that includes a diverse range of recent network traffic attacks and improves predictive performance [9–13]. This research is conducted based on a recent dataset from the “Canadian Institute for Cybersecurity” (CIC-IDS2018 dataset) that incorporates data traffic recorded throughout a large network region. Furthermore, we propose a systematic technique for detecting antecedent anomalies to security infractions in time series data. The technique is based on two main components: time series quantization and the use of Granger causality test. In time series quantization, the data is transformed into a set of discrete values that represent different levels of activity. This step is necessary for the subsequent analysis using the Granger causality test. This test is a statistical method that is used to identify relationships between variables in time series data. The test is used to determine which variables are likely to contain anomalies that precede security infractions. The result of this analysis is used to rank the variables according to their potential to contain such anomalies. Security analysts can utilize these anomalies to better comprehend the evolution of sophisticated computer assaults as well as to trigger warnings that suggest that an attack is impending.

In this study, a novel approach is presented for the effective early detection of intrusions in cloud computing utilizing time series data. A collaborative features selection method is proposed, which integrates time series analysis techniques based on anomaly detection and stationary and causality tests to address the issue of misleading connections between anomalies and attacks. The method was evaluated using the CSE-CIC-ID2018 dataset. The data preprocessing step was carried out using the IDS-Dataset-Cleaning tool, with the stationarity evaluated using the KPSS test. The data was then resampled into 5-minutes bins, columns with Granger causality less than 0.05 were selected, and anomalies were detected through the application of ADTK. A Granger causality matrix was calculated based on the time range, and anomalies that affected future attack labels were selected.

In addition to the proposed collaborative feature selection method, the study also presents a prediction model based on the Facebook Prophet model to evaluate the effectiveness of the proposed method. The conducted experiments demonstrate that the predictors employed in the prediction model have significantly improved the performance metrics and reduced the complexity and resource usage of the model. The results of the evaluation showed a significant increase in the accuracy of forecasting, a decrease in the total number of input predictors, as well as a decrease in prediction time. The number of predictors was reduced from 70 to 10, leading to improved performance metrics, including MAE, MSE, RMSE, MAPE, MdAPE, and DTW. Training, prediction, and cross-validation times have been reduced to approximately 85%, 15%, and 97%, respectively. Although memory consumption remained approximately the same, the reduction in utilization time resulted in a substantial decrease in resource usage.

This research is motivated by concerns about privacy and security in cloud computing, specifically focusing on intrusion detection. The authors highlight the limitations of current network intrusion detection systems (NIDS) in terms of false positives and propose a novel technique based on time series anomalies utilizing machine learning to effectively detect intrusions in cloud computing. The study also introduces a collaborative features selection model that integrates time series analysis techniques with anomaly detection, stationary, and causality tests to address misleading connections between time series anomalies and attacks. The research aims to improve the efficiency and performance metrics of intrusion detection models while reducing resource usage. The proposed approach has the potential to contribute to the growth of cloud computing by enhancing security and privacy in intrusion detection.

The rest of the paper is structured as follows: Sect. 2 presents the background of this study. Related literature is introduced in Sect. 3. The methods used in this research are presented in Sect. 4. Section 5 introduced the proposed model. This section also discussed the steps followed to build the model of the study. Section 6 introduces the results obtained from this study and discussion of results. The paper concludes and suggests future works in Sect. 7.

Background

The distribution of computer services via the Internet with pay-as-you-go pricing is referred to as cloud computing [14]. While cloud computing offers numerous advantages, such as scalability, adaptability, cost-effectiveness, and availability, it has also raised security issues and concerns. In cloud computing, intrusion detection is the process of detecting and responding to unauthorized access, malicious behavior, and security risks [15]. The vast volume of data that must be processed in real-time is one of the obstacles of intrusion detection in cloud computing [16]. ML algorithms are capable of processing large volumes of data and identifying patterns and anomalies that may indicate security breaches, making them important tools in ensuring the security and reliability of cloud-based systems and applications. By leveraging the power of machine learning, organizations can improve the security and reliability of their cloud-based systems and applications [17].

Intrusion detection system (IDS)

An IDS is a hardware or software program that continually analyzes network or system activity for evidence of unauthorized access, security breaches, or anomalies [18]. The two main types of intrusion detection systems are “Network Intrusion Detection System (NIDS)” and “Host Intrusion Detection System (HID)”, where NID monitors network traffic and HID monitors operating system files. According to the “US National Institute of Standards and Technology (NIST)”, most of the alarms generated by NID are false positives [19].

IDSs are further categorized as signature-based and anomaly-based. Signature-based intrusion detection systems, also known as Misuse Detection Systems [20], recognizes known attacks by comparing them to signatures in its database. On the other side, anomaly-based intrusion detection systems, known as rule-based IDS [21], identify potential threats by analyzing traffic behavior rather than relying on predefined signatures. Despite advancements in intrusion detection algorithms and frameworks for securing large-scale networks, there is a need for IDS to become more accurate and effective in detecting a wider range of intrusions while reducing false alarms [22].

Anomaly detection

Anomaly detection is a data analysis process that involves identifying data instances or patterns that deviate from the expected or normal behavior. This technique is used to identify outliers or abnormalities in a given dataset [2, 23, 24]. Anomaly detection approaches like clustering and predictive analysis offer a practical benefit over supervised learning systems due to the constraints and limitations of labeled datasets [25, 26].

Although anomaly detection has been explored in various domains, there is a need for further research and evaluation in cloud environments [5, 27–30]. The timeliness of detection, rate of change, scale, conciseness, and incident description are all important factors to consider when detecting anomalies in systems [25].

There are three types of anomalies [27, 31]: (1) point anomalies, which are isolated data instances that are distinct from the rest of the data, (2) contextual anomalies, which can take place when data deviates from the norm within a specific context or based on its behavioral features, and (3) collective anomalies, which develop when a group of related data instances deviates from the overall average for the entire dataset. The collective category is commonly used for online or real-time anomaly detection .

The Anomaly Detection Toolkit (ADTK) is a Python-based tool for identifying anomalies in time series data using rule-based and unsupervised approaches. The selection and combination of detection algorithms, feature engineering methods and ensemble methods are crucial for developing an effective anomaly detection model. This toolkit offers a uniform API for various detectors, aggregators, transformers, and pipelines that integrates them to a model. It also includes tools to process and visualize time series data and anomalous incidents [27, 32].

Time series data

Time series data is "a sequence of data points collected over time through repeated observations" [33]. The examination of time series data can enhance our understanding of the behavior of events associated with a specific time instance in different domains. There are various time series toolboxes available, each equipped with specialized interfaces for specific model classes such as ARIMA filtering, deep neural networks, or framework interfaces for standalone time series data modeling applications such as forecasts, feature extraction, annotating, and classifying [34].

The models utilized in Time Series Analysis (TSA) are designed to categorize data, reduce noise in signals, and validate assumptions [35]. However, the process of TSA can be a time-consuming one, requiring frameworks to be scalable, capable of handling multiple time series approaches, automatable, and iterative [35].

Time series forecasting

Time Series Forecasting is the technique of predicting data for a certain future period. Research efforts and advancements in cyber security forecasting and forecasting are not as obvious as attack detection, but they are gaining pace [31]. The obvious next step for intrusion detection is to take a proactive strategy, in which we need to anticipate forthcoming harmful acts so that we can respond to such occurrences before they cause any harm. Although this leads to different formulations, two well-known areas of time series analysis are prediction of occurrences using previous behaviors (forecasting) and identifying unusual instances in relation to data item generalization (anomaly detection) [32].

A continuous model-based approach is illustrated with a time series that represents the amount of attacks on a specific network or computer system over time. Some models support specific model families (for example, ARIMA or neural networks), whilst others support broader forecasting frameworks. Others provide more comprehensive prediction frameworks but interfaces to well-known ML tools are lacking like sci-kit-learn [33]. An open-source toolkit that links current machine learning tools, allowing model building, modification, and assessment, is lacking [34].

Stationarity

Time series data is not the same as cross-sectional data. Time series are compilations of data collected at various points in time. Because order matters and most time series forecasting methods assume stationarity, we must make non-stationary data stationary. The KPSS test is used solely on stationary data in this research.

The "Kwiatkowski–Phillips–Schmidt–Shin (KPSS)" test examines whether a time series data is stationary (around a mean) or non-stationary because of a unit root. The mean and variance of a stationary time series are statistical properties that stay constant across time. The test's null hypothesis is that the data is stationary, whereas the other hypothesis assumes that the data is non-stationary [35].

The linear regression underpins the KPSS test. A series is separated into three components a stationary error, a deterministic trend, and a random walk. If data is stationary, then an interceptor with a fixed element will exist, and the series would be stationary around a certain level. A simplified version without the temporal trend component is employed to determine the amount of stationarity [35].

Features selection

Features selection (FS) and dimension reduction methods are important in machine learning for optimizing prediction inputs and improving prediction accuracy [36]. FS involves selecting relevant features based on selection and stopping criteria, and results are evaluated using evaluation criteria. FS methods like wrapper, filter, and embedded/hybrid techniques are commonly used, with filter methods being preferred due to their independence from classifiers [37]. FS is crucial in cybersecurity for reasons such as selecting appropriate feature subsets for different attack types, accommodating changes in network traffic activity, and enhancing detection accuracy and processing efficiency [37]. Proper FS can improve performance evaluation of anomaly-based intrusion detection systems (IDS) and ensemble detection with multiple forms of categorization can help minimize false alarms [38]. Optimization-based variables and parameter tuning can also be employed to choose the optimal feature subset and enhance IDS performance [39].

Wrapper strategies and embedded methods are techniques used for feature selection in machine learning that involve using ML classifiers [40], but wrapper strategies can increase computing complexity due to repetitive learning and cross-validation [41]. Filter methods, on the other hand, are preferred for FS over the wrapper and embedded methods because they are independent of classifiers [42]. Filter methods are classified into univariate and multivariate approaches, with multivariate approaches being preferred as they minimize the redundancy problem by considering mutual correlations among features [43]. According to a survey [42], filter methods are more widely used (47%) compared to embedded (24%) and wrapper methods (29%).

In Intrusion Detection Systems (IDS), feature selection techniques are categorized into classification-based and clustering-based methods, with classification techniques focusing on choosing the best feature combination for establishing correlations between different class labels [44]. The performance of classification-based models is evaluated by inputting the selected features into a training model independent of any learning algorithm [45]. On the other hand, clustering-based feature selection considers all features in the dataset and produces a cluster organization that can be integrated with the learning model to gradually enhance performance by incorporating the most important features. Performance evaluation in anomaly-based IDS datasets is crucial, and recent datasets should be used to confirm the effectiveness of the suggested methods with new types of attacks. Proper feature selection aids in the learning step of detecting cyberattacks in the testing phase, and employing well-tuned parameters and optimization-based variables can enhance anomaly-based IDS. Ensemble detection with multiple forms of categorization can also improve the detection phase while minimizing false alarm rates [46].

Causality

Granger causality is a statistical test employed to evaluate the correlation between two-time series variables to determine if one variable causes a change in another [47]. The Granger causality test is performed using null and alternative hypotheses, with a related F-test statistic and p-value. Rejecting the null hypothesis suggests evidence of causality between the two variables [48]. A modification of the Granger causality test, called variable-lag Granger causality, allows for causality to impact effects with arbitrary time delays. If the causal relation between variables is understood, causation may be used to determine the root causes of anomalies [49].

Facebook's prophet

Facebook's Prophet, a widely used and effective tool for forecasting problems, developed by Facebook's Core Data Science team, is a time series forecasting package released in 2017 and is one of the most popular forecasting algorithms, with over 20 million downloads. Despite some criticisms of its accuracy, it remains in high demand due to its ease of use, confidence intervals, simple visualizations, and ability to outperform standard statistical procedures like Exponential Smoothing (ETS) and Autoregressive Integrated Moving Average (ARIMA) [50]. The Facebook Prophet model can detect outliers and missing data efficiently, breakdown time series into trends, seasonal, and holiday components, and then fit every component to forecast future patterns and trends [18]. Based on a Generalized Additive Model (GAM), the Prophet is suitable for forecasting time series with clear seasonality and can be applied to outlier detection and missing data management [51].

Facebook Prophet offers several advantages over other options such as SKforecast or traditional forecasting methods. Firstly, Facebook Prophet is designed to be user-friendly, with a simple API that makes it easy to implement and configure forecasting models. It also provides automated handling of missing data, outliers, and trend changes, reducing the need for manual data preprocessing and feature engineering [51].

Additionally, Facebook Prophet offers flexibility through the incorporation of domain-specific knowledge, such as custom seasonalities, holidays, and trend change points. This makes it adaptable to different forecasting scenarios, including those with multiple seasonality patterns or irregular data. Moreover, Facebook Prophet is scalable and can handle large datasets with millions of observations, making it suitable for forecasting tasks in big data settings. Furthermore, Facebook Prophet is robust to outliers and can handle missing data and data with irregular sampling frequencies, making it suitable for real-world datasets with noise or gaps. It has been shown to perform well in many forecasting benchmarks and outperforms traditional methods in many cases, especially for datasets with strong seasonalities and trend changes [51].

Literature review

Anomaly traffic detection methods

In recent years, there has been a growing interest in anomaly traffic detection methods using machine learning (ML) algorithms to identify anomalous patterns in network traffic. Various studies have been conducted to propose different approaches for the efficient detection of network anomalies. Some of the notable studies in this area are highlighted and their findings and limitations are discussed.

Sliding window techniques combined with deep learning, machine learning, and neural networks have been proposed in several studies for anomaly traffic detection. For instance, studies such as [52, 53], and [54] have utilized sliding window techniques along with deep learning, machine learning, and neural networks to detect network anomalies. These studies have shown promising results in terms of detecting anomalous patterns in network traffic. Additionally, [55] proposed the use of Long Short-Term Memory (LSTM) and an automated encoder to extract trustworthy features from variable-length data sequences, while [56] introduced an unsupervised deep autoencoder model for training and learning network properties. These studies highlight the potential of using deep learning techniques for anomaly traffic detection.

The study conducted by [57] presents a new approach called the hierarchical adversarial attack (HAA) generation method for targeting intrusion detection systems based on graph neural networks (GNNs) in IoT environments. The method uses a shadow GNN model and a saliency map technique to generate adversarial examples with minimal changes to critical features. A hierarchical node selection algorithm based on random walk with restart (RWR) is employed to identify vulnerable nodes with high attack priority. Evaluation using the UNSW-SOSR2019 data set and comparison with three baseline methods demonstrate that the proposed HAA generation method significantly reduces the classification precision of GNN models, GCN and JK-Net, by more than 30% for NIDS in IoT environments.

Another study [58] proposes a few-shot learning model, called FSL-SCNN, to address the challenge of intelligent anomaly detection in industrial cyber-physical systems (CPS) with limited labeled data. The model uses a Siamese convolutional neural network (CNN) to measure distances between input samples based on optimized feature representations and incorporates a robust cost function design with three specific losses to enhance the training process. Experimental results using fully labeled public datasets and a few labeled datasets demonstrate that the proposed FSL-SCNN significantly improves false alarm rate (FAR) and F1 scores for detecting intrusion signals in industrial CPS security protection.

In addition, the study conducted by [59] highlights the challenges of detecting anomalies in dynamic intrusion detection data, such as multi-aspect data, point anomalies, and group anomalies with attribute correlations. To address these challenges, a novel approach called MDS_AD was proposed, which combines LSH, isolation forest, and PCA techniques. MDS_AD has several key properties, including the ability to operate on multi-aspect data, effectively detect group anomalies, reduce dimensionality with PCA, and perform streaming model updates. Experiments on the UNSW-NB15 dataset confirm that MDS_AD outperforms existing approaches.

The study [60] proposed an optimized intra/inter-class-structure-based variational few-shot learning (OICS-VFSL) model for microservice-oriented intrusion detection in distributed IoT systems. The OICS-VFSL model addresses the challenges of limited computing capabilities of edge devices, imbalanced datasets, and limited computing resources. It utilizes a VFSL framework with an intra/inter-class optimization scheme using reconstructed feature embeddings and introduces an intelligent intrusion detection algorithm for improved multiclass classification. Evaluation experiments show the effectiveness of the proposed model in detecting novel attacks with imbalanced data, outperforming four baseline methods.

Furthermore, some studies have proposed supervised LSTM-based intrusion detection systems for detecting specific types of attacks. For example, [61] proposed a supervised LSTM-based intrusion detection system specifically designed to detect Denial of Service (DoS) and probing attacks. Another study by [62] introduced a deep learning-based parallel cross-convolution neural network for network anomaly detection. These studies emphasize the importance of utilizing specific machine learning techniques tailored to the type of network anomalies being detected.

However, one common challenge in anomaly traffic detection is the high dimensionality of the data, which often leads to a low detection rate and a high false alarm rate. Most of the studies discussed above have utilized datasets such as KDD99, NSL-KDD, UNSW-NB15, and CICIDS2017, which are outdated and have limited features, affecting the efficiency of feature selection. Modern network engines generate far more signals per second, requiring more features for efficient detection [45, 63]. Thus, there is a need for further research to develop methods that can effectively handle high-dimensional data and accurately detect network anomalies in real time [62].

In summary, anomaly traffic detection using machine learning algorithms has gained significant attention in recent years. Studies have proposed various approaches, including sliding window techniques, deep learning, machine learning, neural networks, and specific types of algorithms such as LSTM-based intrusion detection systems. However, challenges such as high-dimensional data and outdated datasets with limited features need to be addressed to improve the accuracy and efficiency of network anomaly detection [64]. Further research is needed to develop robust and scalable methods that can effectively detect anomalies in modern network environments [62].

Features selection methods

The task of intrusion detection in cloud computing presents unique challenges, including dealing with unlabelled datasets, unknown types of malicious traffic, and analyzing time-varying data streams [45]. Directly applying machine learning (ML) to network traffic analysis may not always be practical due to the diverse nature of traffic types, which makes it difficult to develop effective algorithms [46]. To address these challenges and improve the performance of existing ML systems, recent studies have investigated various feature selection techniques.

One approach to feature selection is the use of wrapping strategies, as proposed in some studies [46]. These strategies involve iteratively selecting subsets of features and training ML models to evaluate their performance. The subsets of features that result in the best model performance are then selected. This approach takes into consideration the interactions between features and their impact on model performance, which can lead to more effective feature selection.

Another approach is the use of filter-based methods that utilize feed-forward deep neural networks (FFDNNs) and convolutional neural networks (CNNs) [60]. Filter-based methods involve applying statistical or information-theoretic measures to rank features based on their relevance to the target variable. Features that are deemed most relevant are then selected for model training. The advantage of filter-based methods is their ability to handle large datasets and identify features with high predictive power, which can lead to improved model performance.

However, evaluating the efficacy of feature selection techniques can be challenging, as many studies in intrusion detection rely on datasets with limited features, such as UNSW-NB15 [65], which may not fully represent the complexities of real-world network traffic. This limitation can affect the generalizability and robustness of the proposed methods [65–67]. Despite this limitation, recent studies have made efforts to utilize more recent and comprehensive datasets, such as the Anomaly Detection Toolkit (ATDK) and Stationary Test, for identifying effective features for detecting and predicting attacks [68, 69].

Furthermore, some studies have proposed novel approaches for intrusion detection using deep learning networks. For example, some studies have focused on predicting multivariate cyber risks using deep learning networks, which can provide insights into the relationships between different cyber threats [68]. Other studies have identified TCP flooding DDoS attacks based on causal behavior in network traffic or causal reasoning frameworks [69, 70]. These studies have shown promising results in terms of detecting and predicting attacks using advanced techniques, but they also highlight the challenge of interpretability and understanding the underlying relationships between different cyber threats, as deep learning models can often result in a “black box” understanding [68].

In conclusion, feature selection methods play a crucial role in intrusion detection in cloud computing based on time series anomalies utilizing machine learning. Recent studies have explored different approaches, including wrapping strategies and filter-based methods, to identify relevant features for improving the performance of ML systems. However, evaluating the efficacy of these methods can be challenging due to limited feature datasets and the use of deep learning models, which may lack interpretability [65]. Further research is needed to develop robust and interpretable feature selection techniques that can effectively handle the complexities of network traffic data and improve the accuracy and efficiency of intrusion detection in cloud computing environments [66, 67, 71].

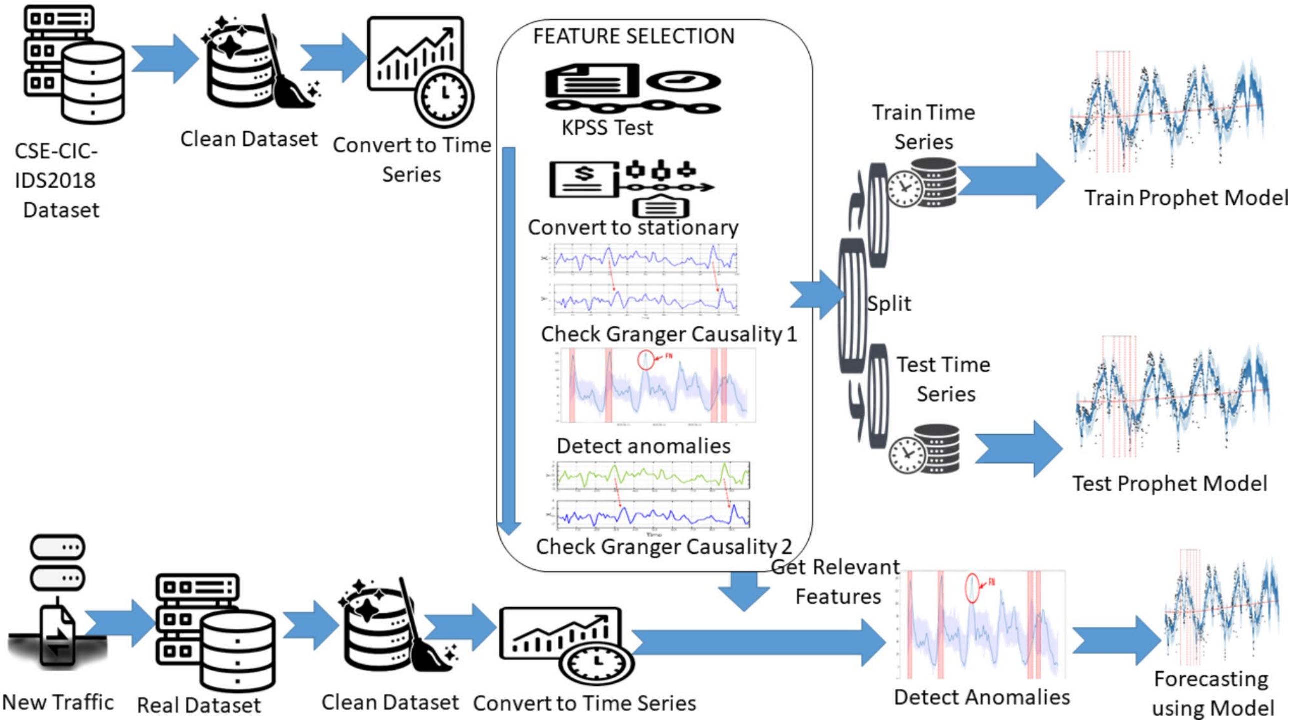

Fig. 1 Research Framework

Research design

This paper proposes a framework consisting of eight steps, as illustrated in Fig. 1, to effectively build and deploy a predictive model for detecting anomalies and forecasting the future behavior of a cloud computing environment.

The framework includes the following essential steps:

First: Selecting the dataset to be used for the model and cleaning it by removing any missing or inconsistent values.

Second: Converting the dataset to time series format, ensuring that it is sorted chronologically.

Third: Using the feature selection technique and selecting the most relevant predictors for the model.

Fourth: Splitting the dataset into training and testing sets to train and validate the model.

Fifth: Training the prophet model on the training dataset.

Sixth: Test the prophet model on the testing dataset to evaluate its performance.

Seventh: Deploy the trained model to detect anomalies and forecast future behavior in real-time on new traffic or a real-world dataset.

Eighth: Continuously monitor the system for any new anomalies, retraining and refining the model as needed to ensure optimal performance.

By following these steps, the proposed framework can effectively address concerns about privacy and security in cloud computing environments. The selected dataset is cleaned and converted to time series format, while the feature selection technique helps to select the most relevant predictors for the model. The dataset is split into training and testing sets to ensure that the model is validated and trained appropriately. The prophet model is then trained and tested, and once its performance is evaluated, it can be deployed to detect anomalies and forecast future behavior in real-time. The model is continuously monitored to ensure optimal performance and to provide early warning of any potential threats, allowing for quick action to be taken to address them.

Methodology

The approach taken in this research is crucial for ensuring the validity and reliability of the results. This section outlines the steps taken for the dataset selection and preprocessing, as well as the metrics used to assess the effectiveness of the intrusion detection system.

Dataset and preprocessing

The availability of datasets is a major challenge in the field of IDS. Although there are various public datasets described in previous research, they have limitations such as being outdated or having redundant records. Some researchers are using self-generated datasets to overcome these issues [72]. However, many of these datasets are anonymized and do not reflect current trends or have desirable statistical properties, meaning a perfect dataset does not exist yet. As network behavior and intrusion methods change, it's important to move away from static datasets and toward dynamic, adaptable datasets that reflect current traffic compositions and intrusions [73].

The inclusion of timestamps for each network flow is crucial for the realism of a dataset. Data cleaning procedures should be fully documented in a research report, including information on any removed or modified rows and columns. The lack of proper data-cleaning information can make it challenging for other researchers to replicate the experiment. Another concern pertains to outdated datasets, such as "KDD 1999", "NSL-KDD", and "ISCX2012". Moreover, researchers may not fully understand the potential impact of reported vulnerabilities on investigations [8]. To address these issues, we use the dataset (CSE-CIC-IDS2018) introduced by the "Canadian Institute for Cybersecurity (CIC)" which is recent and is the most realistic cyber dataset in 2018. The CICIDS2018 dataset is well-suited for classification and detecting time series anomalies in cloud computing environments using machine learning techniques. This is due to its realistic and comprehensive data, established benchmark status, availability of labeled data, and dynamic and diverse characteristics. The dataset contains diverse network traffic data, including various types of attacks commonly found in real-world cloud computing environments, making it a reliable choice for evaluating intrusion detection methods. Additionally, the availability of labeled data enables the use of supervised machine learning techniques, such as classification, for developing accurate and reliable intrusion detection methods [74].

Furthermore, the CICIDS2018 dataset has been widely used as a benchmark dataset in the research community, making it a trusted choice for evaluating the effectiveness of novel intrusion detection approaches in cloud computing environments. The dataset's dynamic and diverse characteristics, reflecting the complexity of real-world cloud environments, allow for the evaluation of intrusion detection methods in detecting time series anomalies under changing network conditions and various types of attacks. Overall, the CICIDS2018 dataset's features make it a suitable choice for classification and detecting time series anomalies in cloud computing environments using machine learning techniques [74].

The CICIDS2018 dataset includes captures and arrangements of machine traffic and framework logs, as well as 80 features selected from the traffic captured using the CICFlowMeter-V3 traffic flow generator. This tool allows for precise customization of characteristics and time flow length. The dataset is in CSV format, with six main features labeled as SourceIP, SourcePort, DestinationIP, DestinationPort, FlowID, and 80 attributes labeled as Protocol [36]. The "CSE-CIC-IDS2018 dataset" was created to support the development of prediction algorithms for network-based intrusion detection.

Feature selection is a crucial part of enhancing the performance of an IDS [74]. The objective is to select a limited number of features to categorize network data, which can not only reduce training time but also improve accuracy by removing irrelevant information. Our approach to feature selection involved three steps. First, we applied the KPA test to select stationary columns from the dataset and converted others to stationary. Second, we included only columns that had Granger causality with the Label column. Third, we selected columns in which anomalies showed Granger causality with the Label column.

Metrics

The selection of an appropriate error metric for evaluating the success of forecasting algorithms is a subject of intense debate. Statisticians advocate for metrics with strong statistical properties, while practitioners prefer metrics that are user-friendly. Researchers argue that the loss function used to train a model should align with the method used for post-performance measurements [75]. The measures used in this study are presented as follows:

Scale-dependent measures.

The prediction error, denoted as e_t at time t, is defined as the discrepancy between the actual value of the series being predicted (y_t) and the forecasted value provided by the forecasting method used (f_t).

$$e_t = y_t - f_t \tag{1}$$

Prediction error is useful in evaluating bias, but it provides limited information on the accuracy of a model's forecasts. To assess the quality of a model's predictions, forecasting accuracy is computed by aggregating individual errors over time horizons using various metrics. The most widely used scale-dependent measures are "Mean Absolute Error (*MAE*)", "Mean Squared Error (*MSE*)", and "Root Mean Squared Error (*RMSE*)" are defined as follows:

$$MAE = \frac{1}{h}\sum_{i=1}|e_i| = \text{ mean }_{i=1,h}|e_i| \tag{2}$$

$$MSE = \frac{1}{h}\sum_{i=1}e_i^2 = \text{ mean }_{i=\overline{1,h}}e_i^2 \tag{3}$$

$$RMSE = \sqrt{\frac{1}{h}\sum_{i=1}^{h}e_i^2} = \sqrt{\text{ mean }_{i=1,h}e_i^2}, \tag{4}$$

where h represents the predicted horizon, lower values indicate greater accuracy [76].

Measures based on percentage errors.

Such measures are used to make the errors scale-independent, it is common practice to represent the error measurements as follows:

$$\text{MAPE } = \frac{1}{\mathrm{h}}\sum_{\mathrm{i=1}}^{\mathrm{h}}|pi| = \text{ mean }_{\mathrm{i}=\overline{1,\mathrm{h}}}|pi| \tag{5}$$

$$MdAPE = \text{ median }_{\mathrm{i}=\overline{1,\mathrm{h}}}|p_i|\,, \tag{6}$$

MAPE stands for "Mean Absolute Percentage Error" and MdAPE is the "Median Absolute Percentage Error". In addition to facilitating the assessment of Improving the accuracy of forecasting for multiple time series of varying magnitudes, these metrics have the advantage of being easily understandable, especially within organizations, with MAPE being the most widely used option [70].

Time was used as a metric to evaluate attack prediction systems by researchers. They assessed the accuracy of predicting the time of the attack by comparing the expected time with the forecasted time [76–79]. This not only predicts when an attack will occur but also provides time for preparation. Practitioners need to have time to respond, so it is a practical way to evaluate prediction methods. It is also significant in evaluating a prediction system.

Dynamic Time Warping (DTW).

Dynamic Time Warping (DTW) is a widely used method for comparing and understanding Time Series data. It is a distance-based approach that enables the comparison of two or more Time Series, while ignoring variations in speed and shift, by aligning them on the time axis. The method calculates the similarity score by finding the minimum distance between corresponding points of two Time Series, after allowing for some amount of warping or stretching of the time axis. DTW effectively handles Time Series data with similar shapes but different phases, by considering the amplitudes at surrounding time points. The comparison is typically done using Euclidean distance or other similarity measures.

Mathematically, DTW can be defined as follows: given two Time Series X = {x1, x2, ..., xn} and Y = {y1, y2, ..., ym}, DTW computes the distance between the two Time Series as the minimum cumulative distance between corresponding points of the Time Series, subject to a warping constraint. The cumulative distance matrix D is computed using a recurrence relation, as shown below:

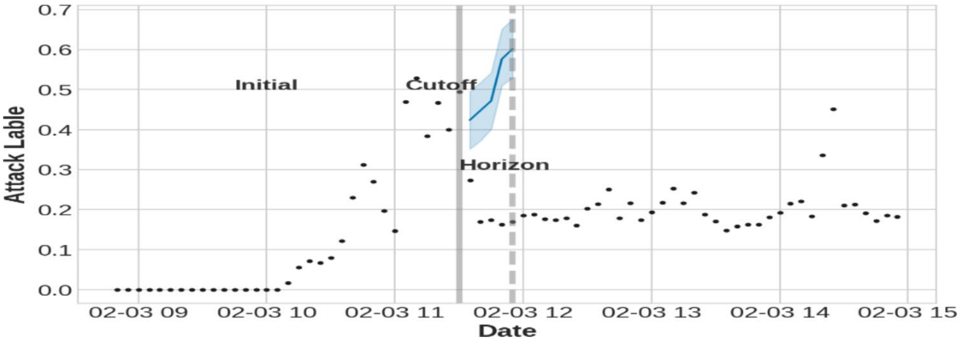

Fig. 2 Prophet's cross-validation process

$$\mathrm{D(i,\ j) = d(xi,\ yj) + min(D(i-1,\ j),}\\ \mathrm{D(i,\ j-1),\ D(i-1,\ j-1))} \tag{7}$$

where d(xi, yj) is the distance between the i-th point of Time Series X and the j-th point of Time Series Y. The warping constraint ensures that corresponding points are aligned on the time axis, and it is typically defined as |i-j| <= r, where r is a user-defined parameter that determines the amount of warping allowed.

The method of FastDTW is a simplified and efficient approximation of DTW. It operates by projecting and transforming a distorted path from a lower resolution to the current resolution using a multi-level approach. FastDTW is capable of processing significantly larger datasets compared to DTW due to its linear time and space complexity. It surpasses DTW in terms of speed and can be effectively utilized in combination with other indexing techniques to enhance time series similarity search and classification (47).

Evaluation of performance

Cross-validation for time series might be difficult at times, but the Prophet technique eliminates the need to write your own function for a rolling forecast. We utilize the cross-validation feature, where the parameters to specify are the forecast horizon and, optionally, the initial training period's size and the spacing between cutoff dates.

Cross-validation is being used with the Prophet model. The first 120 min of data will be used to train the first model. It will forecast data for the next 25 min (because the horizon is set to 25). The model will then train on the initial period plus the period (in this example, 120+25 min) and estimate the following 25 min. It will continue in this manner, adding another 25 min to the training data and then predicting for the following 25 min until there is no data to do so. Figure 2 shows the Prophet's cross-validation process.

The cross-validation output appears as a data frame, as illustrated in Fig. 2. The data frame includes several values at the cutoff point, which marks the final time point in the training set. For instance, "ds" represents the time points in the test set, "y" denotes the actual value at "ds," and "yhat-lower" and "yhat-upper" are the lower and upper bounds of the confidence interval, respectively. On the other hand; the object generated by cross-validation contains more data. This may be described by aggregating to improve performance metrics. For each horizon, the data which obtained from the cross-validation objects are pooled and various metrics are computed.

The performance metrics tool can furnish useful statistics regarding the prediction performance concerning the distance from the cutoff (yhat, yhat-lower, and yhat-upper with respect to y) or the forecast time. After sorting by the horizon, they are generated on a rolling window of forecasts. The statistical estimations are MSE, RMSE, MAE, MAPE, and MdAPE.

The proposed model

This section provides a summary of the steps taken to develop the study's model.

First step The CSE-CIC-IDS2018 dataset was selected for evaluating our technique as it is the most recent and practical cyber dataset widely used for intrusion detection and malware prediction. Choosing this dataset provides a systematic approach for the development of diverse and comprehensive benchmark datasets for intrusion detection. The Botnet attacks is the subset chosen in our experiment.

Second step The IDS-Dataset-Cleaning Tool [80] was used to prepare and preprocess the dataset. This involved eliminating dataset-specific information, dropping features with little variation, deleting duplicate columns, renaming columns, and more. To extract the date and time in the format of Pandas, we parsed the Timestamp field. We also made the Label column categorical and parsed columns to the correct type, removed erroneous data rows, and deleted duplicate rows. After that The Pandas package's read_parquet function was used to import the dataset into Google Collab and the "Timestamp" column

Table 1 Stationary columns based on KPSS test

Destination Port	Bwd Packet Length Std	Packet Length Max	Fwd Act Data Packets
Flow Duration	Flow IAT Max	Down/Up Ratio	Active Mean
Total Fwd Packets	Fwd IAT Total	Subflow Fwd Packets	Active Min
Total Backward Packets	Fwd IAT Max	Subflow Fwd Bytes	Idle Mean
Fwd Packets Length Total	Bwd IAT Min	Subflow Bwd Packets	Idle Std
Bwd Packets Length Total	Fwd Header Length	Subflow Bwd Bytes	Idle Max
Bwd Packet Length Max	Bwd Header Length	Init Bwd Win Bytes	Idle Min

Table 2 The ten stationarity columns which demonstrate a statistically significant Granger causality

'Destination Port'	'Total Fwd Packets'	'Bwd Packet Length Std'	'Flow IAT Min'	'Fwd Header Length'
'Flow IAT Min'	'Subflow Fwd Packets'	'Bwd IAT Total'	'Bwd IAT Std'	'Down/Up Ratio'

was set as the index to make the data frame a time series data. Then the LabelEncoder function from the scikit-learn library is utilized to encode the target labels, benign or malicious traffic, into values ranging from zero to one. Finally, the dataset was divided by time using the Pandas library's loc function, with data before 2 PM considered as training data and data after that considered as test data.

Third step To increase efficiency and accuracy in prediction, the data was evaluated for stationarity using the KPSS test from the statsmodels package. The KPSS test was used to examine stationary columns and develop a method to pick only stationary columns and remove other columns from the data series because stationary data per-

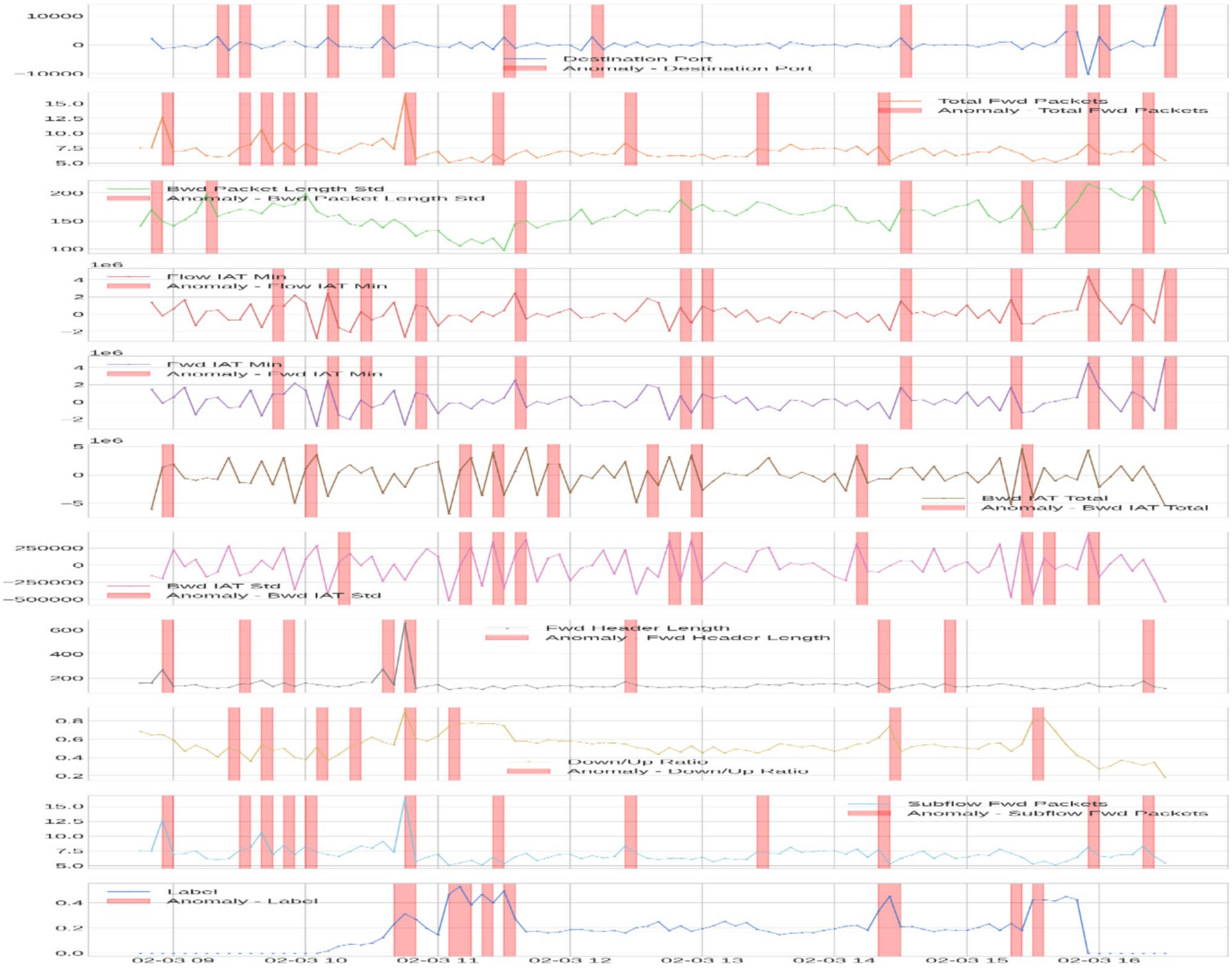

Fig. 3 The anomalies detected in the training dataset

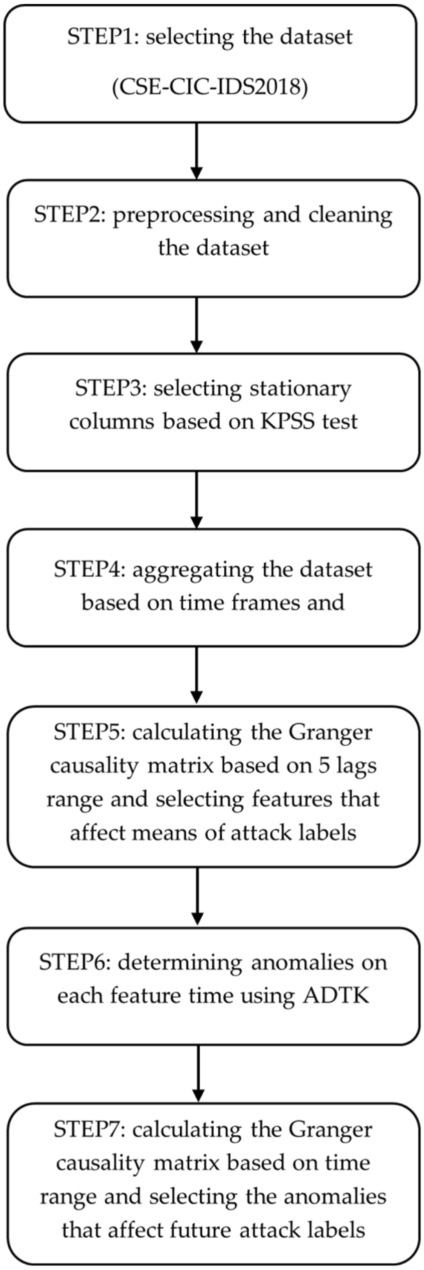
STEP1: selecting the dataset
(CSE-CIC-IDS2018)
STEP2: preprocessing and cleaning the dataset
STEP3: selecting stationary columns based on KPSS test
STEP4: aggregating the dataset based on time frames and
STEP5: calculating the Granger causality matrix based on 5 lags range and selecting features that affect means of attack labels
STEP6: determining anomalies on each feature time using ADTK
STEP7: calculating the Granger causality matrix based on time range and selecting the anomalies that affect future attack labels

Fig. 4 The steps followed to develop the Features Selection model

forms better according to the Prophet algorithm. Table 1 shows the stationary columns based on KPSS test.

Fourth step To resample the data to 5-minute bins, the Pandas data frame's resample() function was used and the mean was determined for all features in the time series. The resample() function divided the DatetimeIndex into time bins and organized the data by these bins. The resample() function returns a Resampler object, similar to a pandas GroupBy object, and an aggregation technique such as mean(), median(), sum(), etc. applied to the grouped data. The Pandas' ffill() function was then used to replace the null values in the time series.

Fifth step Using the Label column values as an indicator of attack existence, the Granger causality test was applied to columns in Table 1 to evaluate if the data in the columns had a causal effect. Only columns with Granger causality less than 0.05 in 5 lags were chosen, equivalent to a 25-minute period. The remaining columns were ten which are listed in Table 2.

Sixth Step To identify outliers and anomalies in the columns resulting from the eighth step, we employed the PersistAD function of the Anomaly Detection Toolkit (ADTK). This function compares each value in the time series to its preceding values using the DoubleRollingAggregate transformer, which is implemented as a pipeline within the PersistAD. The anomalies detected through this method are depicted in Fig. 3 for our training dataset.

Seventh step We conducted another Granger causality test on the columns using the anomalies detected in the previous step to evaluate if the anomaly data has a causal relationship with the attack existence, as indicated by the Label column values. Only columns with Granger causality less than 0.05 over a 5-lag period (equivalent to 25 min) are selected. The remaining columns that resulted after this step are the same as Table 1. Then we used the cross-validation technique (known as rotation estimation or out-of-sample testing) which refers to a group of similar model validation processes used to determine how well a statistical study's conclusions would transfer to a new data set. It is generally used for prediction to see how well a predictive model would function in practice. The above steps are summarized in Fig. 4.

Experiment and results

The Prophet forecasting model was created by following the steps outlined in the previous section. The model was built with lag predictors to broadcast label column values which are indicative of bot assaults. The model was tested on validation data with predictors for all features and with predictors for the selected features. This step was done to determine if an attack can be detected based on earlier abnormalities in the multi-variance time series. Moreover, the testing step can find any impacts on time, memory, and performance. Facebook's prophet algorithm can assess outliers and missing data automatically. In addition, the algorithm deconstructs time series into trend, seasonal, and holiday components, and then fits these individual components to predict the time series' future trend. The Prophet forecasting model provides a built-in tool for visualizing the prediction in the context of the training dataset and creates a plot of the data and

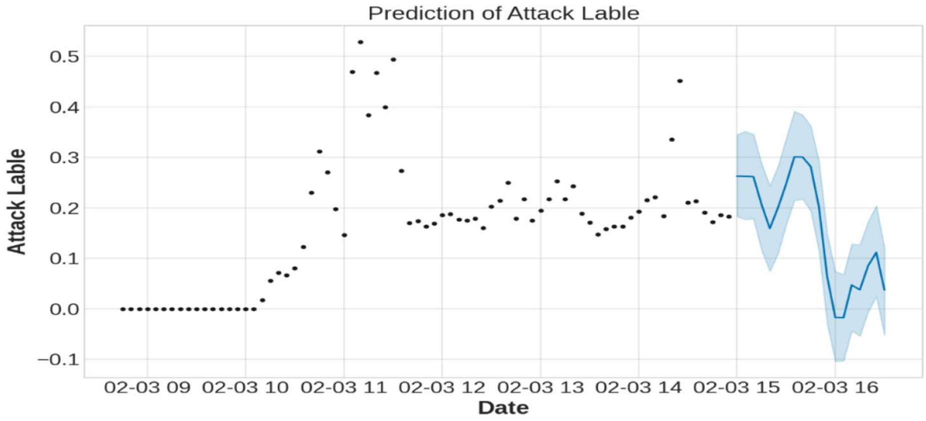

Fig. 5 Results of forecasting the test data using Prophet model

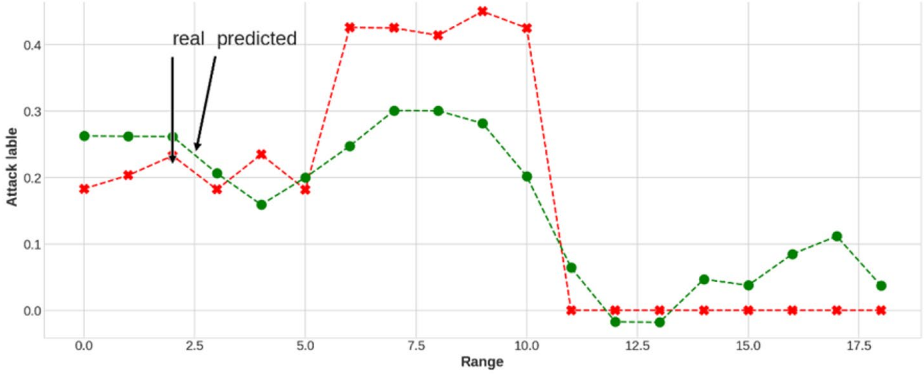

Fig. 6 Comparison between the predicted attack label and the real attack label

overlays the prediction with the upper and lower bounds for the forecast dates. Figure 5 shows the results of forecasting the test data based on the training data in our dataset.

The *MAE* for the training dataset is 0.03 and the *RMSE* is 0.037, whereas the *MAE* for the validation dataset is 0.06 and the *RMSE* is 0.086. This finding suggests that previous anomalies might be utilized as early warning indications for Bot attacks.

Finally, a comparison between the actual and projected values is constructed. Figure 6 shows the results of the comparison which shows a good fit of the model and a reasonable forecast.

For evaluating the performance, cross-validation metrics are compared with and without our Features Selection method. The outcomes of the comparison are illustrated in Fig. 7. The dots reflect the metric value for each forecast, while the line indicates the metric value calculated by averaging the dots in a rolling window. In most cases, we find that forecasts made 5 min ahead of time are more accurate, and that mistakes rise as the horizon is raised for the Prophet model with the Features Selection model. Furthermore, the optimal horizon for the Prophet model alone is 10 min ahead. However, as seen in Fig. 7, the model with Features Selection generates much more accurate results.

To measure time, we employ the ipython-autotime module. Furthermore, we employ memory-profiler, a Python tool that allows for monitoring a process's memory usage, including line-by-line memory consumption analysis of Python scripts. This module employs the psutil package. The outcomes of the experiments are summarized in Table 3.

The outcomes of this study, as indicated in Table 3, demonstrate that the predictors employed in our prediction model have been reduced from 70 to 10, reducing 60 predictors and have significantly improved the performance metrics (MAE MSE, RMSE, MAPE, MdAPE, and DTW) which reduces the error prediction and in turn improves the forecast accuracy. Furthermore, training and prediction times have been reduced. Training time is reduced from 7.37 s to 1 s. Prediction time is reduced from 2.39 to 2.05. Cross-validation time is reduced considerably from 907.8 s to only 21 s. Memory consumption appears to be approximately the same, but we utilize

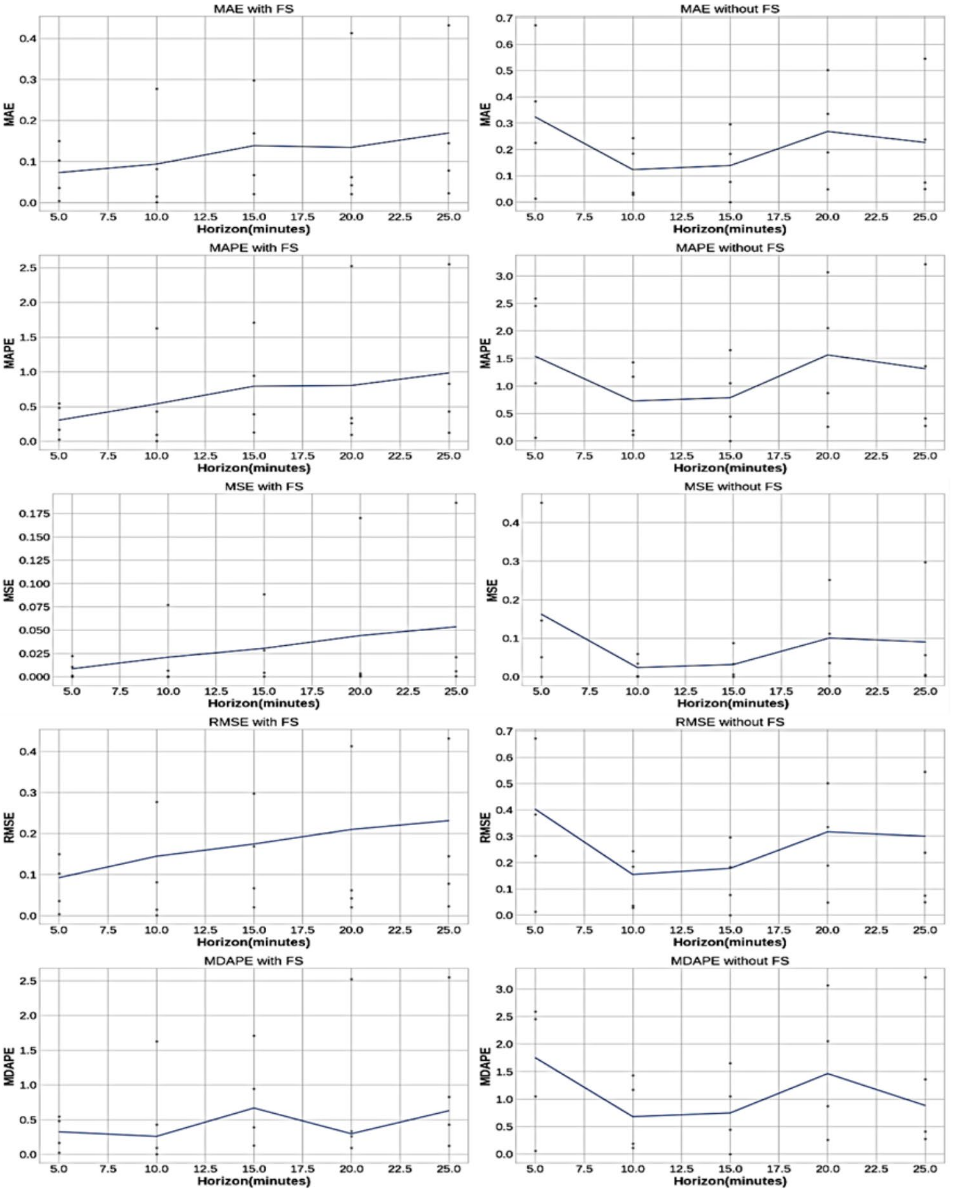

Fig. 7 Comparison between the cross-validation metrics with and without the FS method

Table 3 Summary of experiments results

Model	Predictors	MAE	MSE	RMSE	MAPE	MDAPE	Train time	Predict time	Cross-validation time	Code memory	Cross-validation memory	Distance DTW
Prophet	70	0.216	0.081	0.270	1.188	1.107	7.37 s	2.39 s	907.8 s	500,723	63,928,333	4.063
Prophet with FS	10	0.122	0.032	0.171	0.685	0.436	1 s	2.05 s	21 s	457,118	63,895,858	2.209
DIFF	60	0.906	0.051	0.101	0.503	0.671	6.3 s	0.34 s	886.8 s	43,605	32,475	1.854

it for much less time, resulting in a substantial reduction in resource usage.

The findings of this study show that there are anomalies in the stationary columns that are extracted from the "network traffic generator" utilizing a CICFlowMeter before and during the attack. Because these anomalies have Granger causality with the attack label columns over a 5-lag period, we may consider them as indicators of an approaching attack. We demonstrated how to utilize Facebook Prophet to build a prediction model that communicates attack likelihood before or shortly after it occurs. We use fewer predictors and features to detect attacks in real-time with fewer false alerts.

Conclusion and future works

The field of time series forecasting has progressed significantly over the past few decades. Despite this, concerns must be addressed to enhance prediction accuracy. These include dealing with high-dimensional data and determining the most relevant predictors. Feature selection, a crucial preprocessing phase, has not received much attention. This study presented a collaborative Features Selection method to achieve efficient intrusion detection in cloud networks with time series data. The method integrates anomaly detection and causality tests to distinguish between correlation and causation. The proposed method improved the forecast accuracy by reducing the number of input predictors and saving training time and resources. The experiments were conducted on a large and recent dataset (i.e., CSE-CIC-IDS2018). The experiments have demonstrated encouraging outcomes by attaining an improvement in forecast accuracy, reduction in the number of input predictors, reduction in complexity, and prediction time. The Granger causality metric was used in this study; however, it has been extended to include dissimilarity measures. The proposed model produces decent results and may be improved by tweaking hyperparameters and adding more data into training. In future work, advanced experiments will be conducted to deepen the analysis of causal relationships, generalize the methodology to other prediction models, investigate the role of Features Selection based on different model types, and to compare other prediction models based on our proposed Features Selection method.

Author contributions

Abdel-Rahman Al-Ghuwairi: Conceptualization, Methodology, SupervisionYousef Sharrab: Writing- Original draft preparation, Writing- Reviewing and EditingDimah Al-Fraihat: Writing- Original draft preparation, Writing- Reviewing and EditingMajed AlElaimat: Software, Formal Analysis, VisualizationAyoub Alsarhan: Visualization, Validation, InvestigationAbdulmohsen Algarni: Writing- Reviewing and Editing, Administration, and Funding.

Funding

This research was funded by the Deanship of Scientific Research at King Khalid University, grant number R.G.P.1/188/41.

Data Availability

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

Declarations

Competing interests

The authors declare no competing interests.

Ethical approval

Not applicable.

Received: 15 February 2023 / Accepted: 20 July 2023

Published online: 29 August 2023

References

1. Singh S, Saxena K, Khan Z (2014) Intrusion detection based on artificial intelligence techniques. Int J Comput Sci Trends Technol 2(4):31–35
2. Kene SG, Theng DP (2015), February A review on intrusion detection techniques for cloud computing and security challenges. In *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, 227–232, IEEE
3. Heidari A, Jabraeil Jamali MA, Navimipour J, N., Akbarpour S (2020) Internet of things offloading: ongoing issues, opportunities, and future challenges. Int J Commun Syst, 33(14), e4474
4. Gonçalves, F., Ribeiro, B., Gama, O., Santos, A., Costa, A., Dias, B., … Nicolau,M. J. (2019, October). A systematic review on intelligent intrusion detection systems for VANETs. In *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 1–10, IEEE
5. Mahalakshmi G, Sridevi S, Rajaram S (2016), January A survey on forecasting of time series data. In *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)*, 1–8, IEEE
6. Zouhair C, Abghour N, Moussaid K, El Omri A, Rida M (2018) A review of intrusion detection systems in cloud computing. Secur Priv Smart Sens Networks, 253–283
7. Zhao C, Liu X, Zhong S, Shi K, Liao D, Zhong Q (2021) Secure consensus of multi-agent systems with redundant signal and communication interference via distributed dynamic event-triggered control. ISA Trans 112:89–98
8. Leevy JL, Khoshgoftaar TM (2020) A survey and analysis of intrusion detection models based on cse-cic-ids2018 big data. J Big Data 7(1):1–19
9. Reddy GT, Reddy MPK, Lakshmanna K, Kaluri R, Rajput DS, Srivastava G, Baker T (2020) Analysis of dimensionality reduction techniques on big data. IEEE Access 8:54776–54788
10. Almomani O (2020) A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms. Symmetry 12(6):1046
11. Samadi Bonab M, Ghaffari A, Soleimanian Gharehchopogh F, Alemi P (2020) A wrapper-based feature selection for improving performance of intrusion detection systems. Int J Commun Syst, 33(12), e4434
12. Torabi M, Udzir NI, Abdullah MT, Yaakob R (2021) A review on feature selection and ensemble techniques for intrusion detection system. Int J Adv Comput Sci Appl, *12*(5)
13. Di Mauro M, Galatro G, Fortino G, Liotta A (2021) Supervised feature selection techniques in network intrusion detection: a critical review. Eng Appl Artif Intell 101:104216
14. Al-Fraihat D, Alzaidi M, Joy M (2023) Why do consumers adopt smart voice assistants for shopping purposes? A perspective from complexity theory. Intell Syst Appl 18:200230
15. Sharrab YO, Alsmirat M, Hawashin B, Sarhan N (2021) Machine learning-based energy consumption modeling and comparing of H. 264 and Google VP8 encoders. Int J Electr Comput Eng (IJECE) 11(2):1303–1310
16. Alsarhan A, Alauthman M, Alshdaifat EA, Al-Ghuwairi AR, Al-Dubai A (2021) Machine learning-driven optimization for SVM-based intrusion detection system in vehicular ad hoc networks. J Ambient Intell Humaniz Comput, 1–10
17. Alsarhan A, Al-Ghuwairi AR, Almalkawi IT, Alauthman M, Al-Dubai A (2021) Machine learning-driven optimization for intrusion detection in smart vehicular networks. Wireless Pers Commun 117:3129–3152
18. Liao HJ, Lin CHR, Lin YC, Tung KY (2013) Intrusion detection system: a comprehensive review. J Netw Comput Appl 36(1):16–24
19. Albasheer H, Md Siraj M, Mubarakali A, Elsier Tayfour O, Salih S, Hamdan M, Kamarudeen S (2022) Cyber-attack prediction based on network intrusion detection systems for alert correlation techniques: a survey. Sensors 22(4):1494
20. Khraisat A, Gondal I, Vamplew P, Kamruzzaman J, Alazab A (2020) Hybrid intrusion detection system based on the stacking ensemble of c5 decision tree classifier and one class support vector machine. Electronics 9(1):173
21. Garcia-Teodoro P, Diaz-Verdejo J, Maciá-Fernández G, Vázquez E (2009) Anomaly-based network intrusion detection: techniques, systems and challenges. computers & security 28(1–2):18–28
22. Liu Z, Zheng R, Lu W, Xu S (2020) Using event-based method to estimate cybersecurity equilibrium. IEEE/CAA J Automatica Sinica 8(2):455–467
23. Taylor SJ, Letham B (2018) Forecasting at scale. Am Stat 72(1):37–45

24. Pokharel P, Sigdel S, Pokhrel R, Joshi B (2019), November Time Series Based Pattern Recognition for Anomaly Detection from System Audit Logs. In *2019 Artificial Intelligence for Transforming Business and Society (AITB)* (Vol. 1, pp. 1–6). IEEE
25. Raguseo E (2018) Big data technologies: an empirical investigation on their adoption, benefits and risks for companies. Int J Inf Manag 38(1):187–195
26. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv (CSUR) 41(3):1–58
27. Analytics A (2020) Anomaly detection toolkit. URL https://adtk.readthedocs.io/en/stable
28. Ali A, Hamouda W, Uysal M (2015) Next generation M2M cellular networks: challenges and practical considerations. IEEE Commun Mag 53(9):18–24
29. Chatfield C, Xing H (2019) The analysis of time series: an introduction with R. CRC press
30. Jagreet Kaur. Anomaly detection with time series forecasting: Complete guide (2022) https://www.xenonstack.com/blog/time-series-deep-learning
31. Ramaki AA, Atani RE (2016) A survey of IT early warning systems: architectures, challenges, and solutions. Secur Communication Networks 9(17):4751–4776
32. Faniband YP, Shaahid SM (2021) Univariate Time Series Prediction of wind speed with a case study of Yanbu, Saudi Arabia. Int J 10(1):257–264
33. Faniband YP, Ishak I, Sait SM (2022) A review of open source software tools for time series analysis. *arXiv preprint arXiv:2203.05195*
34. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825–2830
35. Brownlee J (2020) How to check if Time Series Data is stationary with Python. [online]. Machine Learning Mastery
36. Hmamouche Y, Casali A, Lakhal L (2017), May A causality based feature selection approach for multivariate time series forecasting. In *DBKDA 2017, The Ninth International Conference on Advances in Databases, Knowledge, and Data Applications*
37. Benaddi H, Ibrahimi K, Benslimane A, Qadir J (2020) A deep reinforcement learning based intrusion detection system (drl-ids) for securing wireless sensor networks and internet of things. In *Wireless Internet: 12th EAI International Conference, WiCON 2019, TaiChung, Taiwan, November 26–27, 2019, Proceedings 12* (73–87). Springer International Publishing
38. De la Hoz E, De La Hoz E, Ortiz A, Ortega J, Martínez-Álvarez A (2014) Feature selection by multi-objective optimisation: application to network anomaly detection by hierarchical self-organising maps. Knowl Based Syst 71:322–338
39. Pushpam CA, Jayanthi JG (2020), July Methodical Survey on IDS with Feature Selection. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, 606–613, IEEE
40. Wang Q, Zhao D, Wang Y, Hou X (2019) Ensemble learning algorithm based on multi-parameters for sleep staging. Med Biol Eng Comput 57:1693–1707
41. Naheed N, Shaheen M, Khan SA, Alawairdhi M, Khan MA (2020) Importance of features selection, attributes selection, challenges and future directions for medical imaging data: a review. Comput Model Eng Sci 125(1):314–344
42. Venkatesh B, Anuradha J (2019) A review of feature selection and its methods. Cybern Inf Technol 19(1):3–26
43. Bhattacharyya DK, Kalita JK (2013) Network anomaly detection: a machine learning perspective. CRC Press
44. Singh A, Singh Y, Singh R (2013) Improving efficiency and accuracy of classification and clustering of a text documents with feature selection. Int J Eng Res Technol (IJERT) 1(2)
45. Oladimeji O, Olayemi BK, Alese AO, Adetunmbi, Aladesote Olomi Isaiah (2020) Evaluation of selected stacked Ensemble Models for the optimal multi-class Cyber-Attacks Detection. Int J Cyber Situational Aware 5(1):26–48
46. Bouzoubaa K, Taher Y, Nsiri B (2022) DOS-DDOS attacks Predicting: performance comparison of the Main feature selection strategies. Int J Eng Trends Technol 70(1):299–312
47. Belkhouja T, Yan Y, Doppa JR (2022) Dynamic time warping based Adversarial Framework for Time-Series Domain. IEEE Trans Pattern Anal Mach Intell.
48. Otneim H, Berentsen GD, Tjøstheim D (2022) Local lead–lag Relationships and Nonlinear Granger causality: an empirical analysis. Entropy 24(3):378
49. Wang Y, Yu Z, Zhu L (2023) Intrusion detection for high-speed railways based on unsupervised anomaly detection models. Appl Intell 53(7):8453–8466
50. Chen Z, Peng Z, Zou X, Sun H (2022), January Deep learning based anomaly detection for muti-dimensional time series: A survey. In *Cyber Security: 18th China Annual Conference, CNCERT 2021, Beijing, China, July 20–21, 2021, Revised Selected Papers*, 71–92. Singapore: Springer Nature Singapore
51. Purwandari T, Zahroh S, Hidayat Y, Sukonob S, Mamat M, Saputra J (2022) Forecasting model of COVID-19 pandemic in Malaysia: an application of time series approach using neural network. Decis Sci Lett 11(1):35–42
52. Wang Z, Guo Y, Montgomery D (2022) Machine learning-based algorithmically generated domain detection. Comput Electr Eng 100:107841
53. Shao N, Chen Y (2022) Abnormal data detection and identification method of distribution internet of things Monitoring Terminal based on Spatiotemporal correlation. Energies 15(6):2151
54. Blanco R, Pedro M, Juan C, José M (2018) *Multiclass network attack classifier using CNN tuned with genetic algorithms*. In 28th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS), 177–182. IEEE
55. Zhou H, Kang L, Pan H, Wei G, Feng Y (2022) An intrusion detection approach based on incremental long short-term memory. Int J Inf Secur, 1–14
56. Ghorbani A, Fakhrahmad SM (2022) A deep learning approach to network intrusion detection using a proposed supervised sparse auto-encoder and svm. Iran J Sci Technol Trans Electr Eng 46(3):829–846
57. Zhou X, Liang W, Li W, Yan K, Shimizu S, Kevin I, Wang K (2021) Hierarchical adversarial attacks against graph-neural-network-based IoT network intrusion detection system. IEEE Internet of Things Journal 9(12):9310–9319
58. Zhou X, Liang W, Shimizu S, Ma J, Jin Q (2020) Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems. IEEE Trans Industr Inf 17(8):5790–5798
59. Qi L, Yang Y, Zhou X, Rafique W, Ma J (2021) Fast anomaly identification based on multiaspect data streams for intelligent intrusion detection toward secure industry 4.0. IEEE Trans Industr Inf 18(9):6503–6511
60. Liang W, Hu Y, Zhou X, Pan Y, Kevin I, Wang K (2021) Variational few-shot learning for microservice-oriented intrusion detection in distributed industrial IoT. IEEE Trans Industr Inf 18(8):5087–5095
61. Wei Y, Wu F (2022) A Self-adaptive Intrusion Detection Model Based on Bi-LSTM-CRF with Historical Access Logs. In *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery: Proceedings of the ICNC-FSKD 2021 17* (pp. 185–197). Springer International Publishing
62. Toma TI, Choi S (2022) A parallel Cross Convolutional recurrent neural network for Automatic Imbalanced ECG Arrhythmia detection with continuous Wavelet Transform. Sensors 22(19):7396
63. Ahmad R, Alsmadi I, Alhamdani W, Tawalbeh LA (2022) Towards building data analytics benchmarks for IoT intrusion detection. Cluster Comput 25(3):2125–2141
64. Salman EH, Taher MA, Hammadi YI, Mahmood OA, Muthanna A, Koucheryavy A (2022) An anomaly intrusion detection for high-density internet of things Wireless Communication Network Based Deep Learning Algorithms. Sensors 23(1):206
65. Kim MS, Shin JH, Hong CS (2022), September Network Intrusion Detection System using 2D Anomaly Detection. In *2022 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 1–4. IEEE
66. Gaber T, El-Ghamry A, Hassanien AE (2022) Injection attack detection using machine learning for smart IoT applications. Phys Communication 52:101685
67. Moustafa N, Slay J (2015) UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *Military Communications and Information Systems Conference (MilCIS)*, Canberra, ACT, Australia, 1–6, doi: https://doi.org/10.1109/MilCIS.2015.7348942
68. Zängerle D, Schiereck D (2022) Modelling and predicting enterprise-level cyber risks in the context of sparse data availability. *The Geneva Papers on Risk and Insurance-Issues and Practice*, 1–29
69. Pietrantuono R, Ficco M, Palmieri F (2023) Testing the resilience of MEC-based IoT applications against resource exhaustion attacks. IEEE Trans Dependable Secur Comput.
70. Zeng Z, Peng W, Zeng D, Zeng C, Chen Y (2022) Intrusion detection framework based on causal reasoning for DDoS. J Inform Secur Appl 65:103124
71. Ali K, Alzaidi M, Al-Fraihat D, Elamir AM (2023) Artificial Intelligence: benefits, application, ethical issues, and organizational responses. Intelligent Sustainable Systems: selected Papers of WorldS4 2022, volume 1. Springer Nature Singapore, Singapore, pp 685–702
72. Pawlicki M, Kozik R, Choraś M (2022) A survey on neural networks for (cyber-) security and (cyber-) security of neural networks. Neurocomputing 500:1075–1087
73. Yusof NNM, Sulaiman NS (2022), August Cyber attack detection dataset: A review. In *Journal of Physics: Conference Series* (Vol. 2319, No. 1, p. 012029). IOP Publishing

74. Mushtaq E, Zameer A, Umer M, Abbasi AA (2022) A two-stage intrusion detection system with auto-encoder and LSTMs. Appl Soft Comput 121:108768
75. Koutsandreas D, Spiliotis E, Petropoulos F, Assimakopoulos V (2022) On the selection of forecasting accuracy measures. J Oper Res Soc 73(5):937–954
76. Albasheer, H., Md Siraj, M., Mubarakali, A., Elsier Tayfour, O., Salih, S., Hamdan,M., … Kamarudeen, S. (2022). Cyber-attack prediction based on network intrusion detection systems for alert correlation techniques: a survey. *Sensors*, *22*(4), 1494
77. Marappan R, Bhaskaran S (2022) Movie recommendation system modeling using machine learning. Int J Math Eng Biol Appl Comput, 12–16
78. Kotenko I, Gaifulina D, Zelichenok I (2022) Systematic literature review of security event correlation methods. IEEE Access.
79. Staroletov S, Chudov R (2022), November An Anomaly Detection and Network Filtering System for Linux Based on Kohonen Maps and Variable-order Markov Chains. In *2022 32nd Conference of Open Innovations Association (FRUCT)* (pp. 280–290). IEEE
80. Miel Verkerken. Miel verkerken / ids dataset cleaning Â· gitlab https://gitlab.ilabt.imec.be/mverkerk/ids-dataset-cleaning

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.