applied sciences

MDPI

Article

An Analysis of Temporal Features in Multivariate Time Series to Forecast Network Events

Soo-Yeon Ji [1,*], Bong Keun Jeong [2] and Dong H. Jeong [3,*]

[1] Department of Computer Science, Bowie State University, Bowie, MD 20715, USA
[2] Department of Management and Decision Sciences, Coastal Carolina University, Conway, SC 29528, USA; bjeong@coastal.edu
[3] Department of Computer Science and Information Technology, University of the District of Columbia, Washington, DC 20759, USA
* Correspondence: sji@bowiestate.edu (S.-Y.J.); djeong@udc.edu (D.H.J.); Tel.: +1-301-860-4458 (S.-Y.J.); +1-202-274-6292 (D.H.J.)

Abstract: Analyzing network traffic over time is crucial for understanding the changes in network activity. To properly examine network traffic patterns over time, multiple network events in each timestamp need to be converted to time series data. In this study, we propose a new approach to transform network traffic data into time series formats by extracting temporal features to analyze normal/attack patterns. The normal patterns indicate network traffic occurred without any intrusion-related activities, whereas the attack patterns denote potential threats that deviate from the normal patterns. To evaluate the features, long short-term memory (LSTM) is applied to forecast multi-step network normal and attack events. Visual analysis is also performed to enhance the understanding of key features in the network. We compared the performance differences using time scales of 60 and 120 s. Upon evaluation, we found that the temporal features extracted with the 60 s time scale exhibited better performance in forecasting future network events.

Keywords: multi-step forecasting; time series; network traffic analysis; wavelet transformation; permutation entropy

Citation: Ji, S.-Y.; Jeong, B.K.; Jeong, D.H. An Analysis of Temporal Features in Multivariate Time Series to Forecast Network Events. *Appl. Sci.* **2023**, *13*, 10411. https://doi.org/10.3390/app131810411

Academic Editors: Konstantinos Rantos, Konstantinos Demertzis and George Drosatos

Received: 27 August 2023
Revised: 13 September 2023
Accepted: 15 September 2023
Published: 18 September 2023

1. Introduction

Time series data is a collection of successive observations that are recorded in chronological order. Time series prediction (TSP) involves analyzing historical time series data to discover patterns and predict future values. TSP has commonly been utilized in various fields such as stock price prediction [1], weather forecasting [2], earthquake prediction [3], river water level forecasting [4], physiological symptoms detection [5], and more. Monitoring and forecasting network events are imperative in network intrusion to understand future attack trends. Network traffic analysis poses a particular challenge due to the ever-changing nature of network activities over time. Time series analysis can play a vital role in pinpointing essential attributes of attack events during the examination of network activities. However, it may not be suitable for analyzing network traffic data directly, given that network traffic events generally occur as a series of consecutive observations within the same timestamp. Furthermore, monitoring sudden changes in the network over time can serve as a key characteristic for identifying attack events.

In modern network environments, an enormous volume of network events is generated within seconds. Consequently, the analysis of network events requires significant effort, particularly when dealing with large numbers of captured network variables. To analyze time series network events data, it is critical to transform the data into time series formats with equal intervals. Previously, researchers used aggregation [6] to analyze a large number of streaming intrusion alerts. However, applying the aggregation technique to time-stamped events may be susceptible to high variation because of numerous events at

time t_i. In this study, we propose an approach to convert multiple sequences of network events to time series formats and forecast network normal and attack events. Specifically, our approach extracts temporal features by utilizing statistical measurements and computational methods to convert network events to time series data with a targeted time scale. A publicly available network traffic dataset captured in a honeypot system is used to test our proposed approach and to highlight the effectiveness of identifying attack patterns and forecasting possible future attacks. To determine the effectiveness of forecasting future attack patterns, we evaluated two different time scales to predict network attack events and understand the temporal patterns of attackers. We also performed a series of data analyses to determine the effectiveness of our proposed approach. In summary, we make the following contributions in this study:

- An advanced approach is introduced to extract temporal features by integrating wavelet transform, permutation entropy, and statistical measurements from network events. We also propose a new way to extract temporal features of categorical network traffic variables.
- The performance evaluation with different time scales ($t_s = 60$ and 120 s) is conducted to determine the effectiveness of the proposed approach.
- Deep learning (DL) is utilized to evaluate the features and forecast multiple outputs (i.e., network normal and attack events).
- Visual analysis with multiple visualization techniques is performed to determine the effectiveness of analyzing network events with the extracted temporal features.

This paper begins by describing related work in Section 2. Then, we explain our proposed approach and the network traffic dataset in Section 3. In Section 4, experimental results are presented. Lastly, we discuss the implications of this study and future research in Sections 5 and 6.

2. Related Work

Capturing the trends from time series data is a major task in analyzing anomaly detection. Researchers have employed a range of methods to improve the performance of network traffic prediction. These methods encompass statistical, machine learning, and deep learning techniques. Also, time series techniques, such as AR, ARMA, and ARIMA, are often used to predict time series data [7].

Wan et al. [8] proposed a predictive model for attack behaviors by aggregating traffic flows into bags, which are fixed intervals, using the distribution of data within each bag. The Gaussian mixture model (GMM) was used to check the data distribution. They analyzed the Kyoto 2006+ dataset and found that the model effectively predicted the number of security incidents. Werner et al. [6] introduced a novel approach known as CLEAR (concept learning for intrusion event aggregation in real time), which involves grouping intrusion alerts through concept learning and matching incoming alerts with attack behaviors that exhibit similar temporal characteristics. The study employed the concept of inter-arrival times (IATs) to aggregate alerts. By conducting a two-sample KS-Test, the approach can generate aggregated alert groups that are statistically similar to check any significant differences with the current aggregates and determine the status of updating known concepts. Yaacob et al. [9] used a univariate autoregressive integrated moving average (ARIMA) method to detect DoS attacks utilizing the protocol information. Zeng et al. [10] introduced a multivariate time series anomaly detection approach based on an adversarial transformer structure to ensure the quality of the Internet of Things (IoT) services. Abdullah et al. [11] proposed a cyber defense system using generalized autoregressive moving average (GARMA) to predict hourly attack rates. This study emphasized the significance of anticipating potential future attacks as such advanced predictions can provide valuable information to system administrators.

Sokol et al. [12] conducted a time series analysis employing the Box-Jenkins technique. They applied this technique to create autoregression (AR) models for predicting attacks, using network traffic information obtained from honeypots. The study suggested that

bootstraps based on AR(p) were appropriate for attack prediction, particularly when $p = 1$, indicating that the number of preceding values used to predict the following values was 1. Lee et al. [13] introduced an algorithm for proactive real-time anomaly detection. This algorithm employed long short-term memory (LSTM) to predict network anomalies in incoming data. Using short-term historical data points, the approach identified anomalies and dynamically adjusted the detection threshold over time. As a result, the algorithm could detect anomalies in real time without human intervention, offering early warnings. However, the approach accumulated time series data which used $\{t_0, t_1, \cdots, t_{n-1}\}$ to generate an LSTM model and predict whether the upcoming at t_n data point showed an anomaly based on a threshold. The proposed RePAD can detect anomalies to provide early warnings in real time. The LSTM model is continuously generated depending on a dynamically calculated threshold. Viinikka et al. [14] presented a time series model to filter out irrelevant alerts from alert flows using alert aggregates. The primary objective of this study was to remove alerts associated with normal flow behavior. A non-stationary autoregression (NAR) model was generated, and a Kalman fixed-lag smoother algorithm was utilized to estimate the parameters for the NAR. The results of the study demonstrated an enhancement in model accuracy through the implementation of the NAR model.

Fouladi et al. [15] proposed a DDoS attack detection approach using an autoregressive integrated moving average (ARIMA) model. The exponential filter and the dynamic threshold method were utilized to identify the changes in the network. For the study, source/destination IP address features were used. The proposed approach exhibited high accuracy and low false alarms. Nezhad et al. [16] used packet and source IP address data to calculate a feature as a one-minute interval time series, aiming to predict DoS and DDoS attacks. Specifically, the number of packets in every following minute was predicted using ARIMA. Ergan et al. [17] introduced a time series analysis to identify anomalies. The study utilized LSTM-based neural networks to find the optimal length for sequence data. Then, one-class support vector machines (OC-SVMs) or a support vector data description (SVDD) was used to determine the anomaly. Salahuddin et al. [18] introduced a method called Chronos, which utilizes a time-based autoencoder technique for identifying DDoS anomaly traffic. This technique aggregates features extracted from packets across different time windows and subsequently compares their respective performances.

Given that a time series comprises sequential data points obtained at consistent time intervals, network traffic data occurring across successive network events should be represented using time series analysis techniques. Previous studies used different approaches such as aggregation [14,19] or subtraction [20] to present time series data. While the studies have incorporated time series analysis, they predominantly employed univariate data to construct time series-based models through data aggregation. Furthermore, limited studies have examined consecutive sequences of observations like network traffic events. Numerous studies have been conducted to forecast short-term and long-term network traffic changes within data center networks. They used various time intervals of minutes, hours, and days to generate predictive models for better resource utilization scenarios [21–23]. However, limited studies have been performed to forecast possible network attacks in the network security community. Since a high volume of network traffic events occurs in seconds and minutes (often caused by DOS or DDOS attacks), finding appropriate time intervals or window sizes to forecast network attacks is considered a research challenge. In our previous work [24], we conducted a study incorporating statistical measurements to represent continuous variables in order to predict attacks by comparing short time scales (1, 5, and 15 s). While the study demonstrated good predictive capabilities for network attacks, such short time scales require high computational complexity because a large volume of data needs to be processed. Increasing time intervals can be an alternative approach to resolving this complexity problem. But, increased time intervals may impact the efficiency of extracted features significantly. Overall, no comparative study has been performed to show optimal time intervals in identifying network attacks due to constant changes in attack types. In electric power cyber-physical systems, researchers

considered each type of network attack sequence as invalid attack results if the time interval exceeds 240 s [25]. However, many studies used 60 s to analyze network traffic data [26–28] to evaluate high bursts of attack events. Therefore, we utilize 60 s intervals to assess the network traffic data. Alternatively, 120 s intervals are used to determine the performance differences.

Also, most previous studies did not take into account categorical variables in network intrusion detection analysis. Given the significance of incorporating categorical variables in intrusion detection analysis, [29], we introduce an advanced approach to represent multiple sequential timestamped events as time series data by extracting temporal features and analyzing categorical variables. We also assess these temporal features using long short-term memory (LSTM) to predict future attacks and normal events.

3. Methodology

3.1. Dataset

We used a publicly available dataset (called the Kyoto dataset [30]). The dataset was generated by capturing real network traffic data within a honeypot environment. The honeypot comprises multiple computer systems that mimic a real computing environment, designed to deceive cyber attackers into perceiving it as a viable target. The rationale behind creating such an environment is to lure attackers and gain insights into their behaviors, as well as comprehend security vulnerabilities. Therefore, all network events coming to the honeypot environment are considered legitimate attacks. The dataset consists of twenty-four variables, including fourteen variables that are identical to the variables presented in the KDD Cup 99 dataset [31]. The dataset contains three distinctive network event categories: normal activities, known attacks, and unknown attacks. Due to the limited amount of unknown attack instances, 'known attacks' is designated as 'attack' within the scope of this study. Also, three trigger variables, including IDS_detection trigger, malware_detection trigger, ashula_detection trigger, were excluded. These three variables represent detected triggering alerts by the IDS system or detection software. They are not considered in the study because they possess limited information to predict normal or attack events.

3.2. Methods

Assume that the given original network sequence is comprised of a series of time sequence observations $\mathcal{O} = \{(T_i, X_i, Y_i)\}, i = 1, 2, \cdots, N$ contains network traffic events (N is the total number of events). T represents time, X indicates the variables in the original network traffic sequences with nominal, real numbers, and binary variables, and $Y \in \{\text{normal, attack}\}$ indicates network events. To identify the underlying patterns of network events over time, the time sequence observation series ($\mathcal{O}$) needs to be transformed into time series data with a regular interval. Our approach consists of three steps: (1) generating one-hot encoded variables, (2) constructing time series with a pre-defined time scale t_s, and (3) generating forecasting models. A performance evaluation was conducted with different time scales ($t_s = 60$ and 120 s) to determine the effectiveness of the proposed approach.

3.2.1. Generating One-Hot Encoded Variables

The original network event data contain three categorical variables: service type, flag, and protocol type. The service type indicates network connection types, such as HTTP, telnet, FTP, etc. The flag represents the state of the network connection. There are about thirteen connection states, including S0, S1, SF, REJ, S2, S3, RSTO, RSTR, RSTOS0, RSTRH, SH, SHR, and OTH, and each state indicates a specific network connection state. For more details about the connection states, please refer to ref. [30]. Protocol type denotes network protocols used by each network connection. One-hot encoding [32] is applied to the nominal variables. This process transforms categorical variables into numerical data by replacing each attribute value with a binary representation (1 or 0), which indicates the position of a corresponding attribute value. For example, the protocol type variable

contains three attribute values: TCP, UDP, and ICMP. By applying one-hot encoding, three one-hot encoded nominal features are generated. Similarly, the remaining categorical variables are converted to one-hot encoded variables.

3.2.2. Constructing Time Series Data with a Pre-Defined Time Scale t_s

Time series data are constructed by mapping a set of network traffic data series in t_s to values over time through the extraction of temporal features. First, the original data is segmented with a pre-defined time scale t_s. A new time index is created depending on the pre-defined time scale t_s over time, $\nabla t_i = (t_s * c) - \nabla t_{i-1}, c = 2, \cdots, t_N, i = 2, \cdots, t_N, \nabla t_1 = t_s$ where $t_N = \frac{m_t}{t_s}$, m_t is a maximum time. Thus, a new time index is generated as $\nabla t_i = \{t_1^{'}, t_2^{'}, \cdots, t_N^{'}\}$. Within each time index, ∇t_i contains a series of tuples $\{(X_i, Y_i)\}$ forming M $\times$ J matrix ($X_i \in R^{M \times J}$) and M $\times$ D matrix ($Y_i \in R^{M \times D}$), where $M(M \geq 1)$ indicates the total number of observations at time ∇t_i, J represents the total number of variables, and $D(D \geq 1)$ denotes the size of dependent variables. It is important to note that the size of M may vary because the number of network events occurring over time is different.

For the one-hot encoded dependent variables, the frequency of network events over ∇t_i is computed as $\mathcal{C}(Y_i^k) = \Sigma_1^{n_t} I_{Y_i}(\delta_i)$, where δ_i indicates if each network event is normal or an attack. The frequency of each one-hot encoded variable over ∇t_i is also measured for the nominal variables. For instance, for the variables (source and destination port numbers), the number of used port numbers over ∇t_i are counted. For the other variables, a representative value for each $M-$dimensional vector over ∇t_i is measured to generate time series data with equal intervals. In this paper, we present a methodology that utilizes wavelet transform (DWT [33]) and permutation entropy (PE [34]) to map values from the original time sequences into time series features. The advantages of employing these techniques (i.e., DWT and PE) include the capability to identify sudden network event changes and to illustrate the trend of network event behaviors over time. DWT is well-suited for analyzing non-stationary data, such as network traffic, in both time and frequency domains. It achieves this by continually decomposing the data into two sub-bands. That is, detail and approximation coefficients are produced by successively passing data through high-pass and low-pass filters until they reach a predefined level. The coefficients represent time and frequency information associated with each decomposition level by the following.

$$\mathcal{W}_c = \quad < d(t), \phi_{(\tau,\gamma)} > \quad = \int_{-\infty}^{\infty} d(t)\phi_{(\tau,\gamma)}dt$$

where $d(t)$ indicates data, $\phi_{(\tau,\gamma)}(t)$ represents a mother wavelet function, and τ and γ denote frequency resolution (i.e., scale) and shift parameters, respectively. Approximation coefficients ($a_{(\tau,\gamma)}$) present low-frequency information, while the detailed coefficients ($d_{(\tau,\gamma)}$) show the high-frequency characteristics of data. PE is applied to analyze the wavelet coefficients. It is a complexity measurement that integrates symbolic patterns and entropy. Specifically, we used the coefficients, $a_{(\tau,\gamma)}$, $d_{(\tau,\gamma)}$, and $a_{(\tau,\gamma)} + d_{(\tau,\gamma)}$ to extract features. PE is used to construct subsequences (s_i) with a pre-defined embedding dimension (e_d). Then, each subsequence is mapped into a unique permutation to capture the order as $\pi(i) = \{0, 1, \cdots, e_d\}$. The probability distribution of the permutation is computed as $p_{\pi(i)} = \frac{\delta_{\pi(i)}}{|s_i|}$, where $\delta_{\pi(i)}$ presents the occurrence of the pattern $\pi(i)$. Lastly, Shannon's rule [35] is utilized to calculate the permutation entropy as $\Sigma_i - p_{\pi(i)} \times log(p_{\pi(i)})$.

In addition, statistical feature ($\mathcal{P}$) is extracted as $\chi(w_{o_i}, o_i)$, where $\chi(\cdot)$ indicates the ANOVA test, w_{o_i} represents the detailed coefficients of o_i, and o_i denotes the ith vector of the original sequences $\mathcal{O}$ at ∇t_i. This feature represents a p-value, indicating if there is any statistical difference between the original sequences and the wavelet coefficients. We also compute an additional feature using the first moment ($\mathcal{E}$) as $\frac{1}{|o_i|}\sum o_i$ in ∇t_i. Algorithm 1 presents a pseudo-code that converts network traffic series to a targeted time scale (t_s).

Algorithm 1: Conversion of network traffic data to time series format with a pre-defined time scale t_s

Input: network traffic observations (O)
Output: converted time series data (F)

for $i = 0$ **to** $\frac{m_t}{t_s}$ **do**
  for $\forall j, v_j \in O_i$ **do**
    /* O_i: i^{th} observations, v_j: jth variable */
    if $v_j \in \mathcal{C}_N$ **then**
      Determine $|\phi(v_j)|$ /* $\phi(\cdot)$: a mapping to determine unique attributes */
      Generate one-hot encoded nominal variables, $n_1, n_2, \cdots, n_{|\phi(\cdot)|}$
      for $k = 1, 2, \cdots, |\phi(\cdot)|$ **do**
        $\mathcal{N}_j[k] = \Sigma I(n_k \in v_j)$ /* $\mathcal{N}$: nominal variable features */
    else if $v_j \in \mathcal{D}_Y$ **then**
      Determine $|\phi(v_j)|$
      Generate one-hot encoded dependent variables, $y_1, y_2, \cdots, y_{|\phi(\cdot)|}$
      for $k = 1, 2, \cdots, |\phi(\cdot)|$ **do**
        $\mathcal{Y}_j[k] = \Sigma I(y_k \in v_j)$ /* $\mathcal{Y}$: dependent variable features */
    else
      /* $\chi(\cdot)$: ANOVA test, $\varphi_w(\cdot)$: wavelet transform, $\psi(\cdot)$: permutation entropy */
      $\mathcal{W}_i[j] = \psi(\varphi_w(v_j))$ /* $\mathcal{W}$: wavelet features */
      $\mathcal{P}_i[j] = \chi(\varphi_w(v_j), X_j)$ /* $\mathcal{P}$: statistical features */
      $\mathcal{E}_i[j] = \frac{1}{|v_j|} \sum v_j$ /* $\mathcal{E}$: first-moment features */
  $\mathcal{F}_i = \{(\nabla t_i, \mathcal{P}_i, \mathcal{E}_i, \mathcal{W}_i, \mathcal{N}_i, \mathcal{Y}_i)\}$

3.3. Generating Forecasting Models

Long short-term memory (LSTM) is used to predict two future outputs (i.e., network normal and attack events). LSTM is a type of recurrent neural network (RNN) architecture that is frequently used in time-series analysis [36]. It addresses the vanishing gradient problem in RNN by providing longer-lived short-term memory to preserve information across timesteps [37]. It includes four components: a memory cell, an input gate, an output gate, and a forget gate. The memory cell serves as an information store, while the gates regulate the flow of that information [38]. We used a traditional LSTM model with four components to forecast network normal and attack events (see Figure 1). It has three layers to extract temporal features from the data. The model was built with two hidden layers, dense and dropout layers, and used the rectified linear unit (ReLU) activation function to forecast network attacks. The mean squared error (MSE) is used for loss function in model training and validation. Adaptive moment estimation (ADAM) is also used as a model optimizer to compute adaptive learning rates. The root mean square error (RMSE) is computed to evaluate forecasting performances.

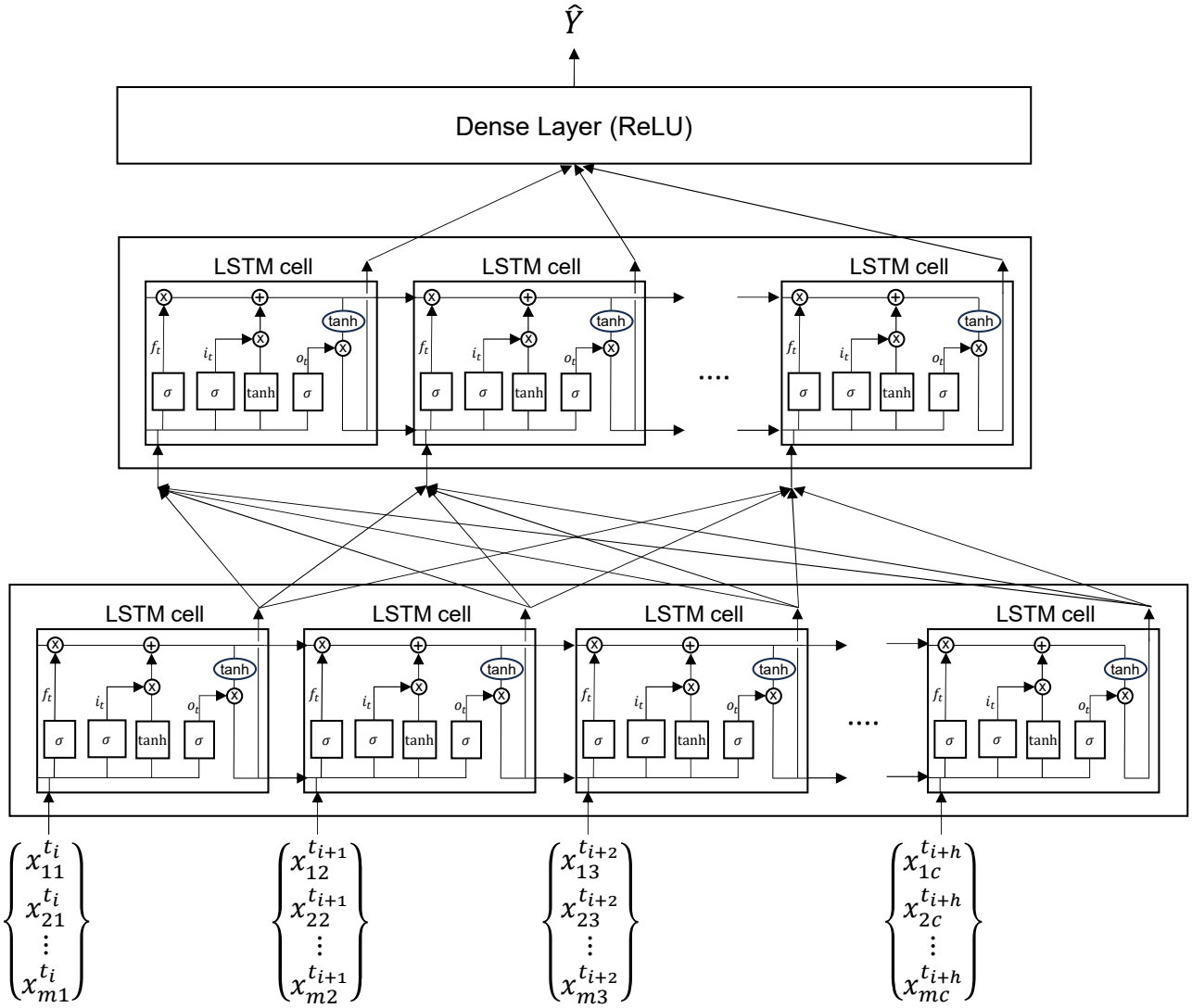


Figure 1. A diagram of the proposed time-series prediction using LSTM.

4. Results

For the data analysis with wavelet transform, we used the decomposition level of three with the 'db3' wavelet function. Different time scales (t_s = 60 and 120 s) and embedding dimensions (e_d = 3 and 4) were taken into account to assess and compare performance differences. Based on the analysis, we also present the results from the forecasting models to predict future network events (i.e., normal and attack events). Rather than employing a one-step forecasting method that predicts a single value based on the past, a multi-step forecasting method is utilized to predict sequences of values.

Figure 2 shows visual representations of variables associated with 'attack' and 'normal' network events on the first day of January 2015. Figure 2a,c present the total number of connections, destination port, and duration using the original network traffic data. The total number of connections represents the number of network connections made in the past two seconds with the same source and destination IP addresses. These variables are used to identify suspicious activities by detecting a significant volume of continuous network connections directed toward the same computer machines. The most commonly used port numbers in cyberattacks are 22—SSH (secure shell), 80—HTTP (hypertext transfer protocol), and 443—HTTPS (hypertext transfer protocol secure). However, by evaluating the destination port information, we found that attackers tried to use various ports to penetrate server machines in a honeypot environment (see Figure 2b). From the analysis

of the duration information, we determined that the time duration for the attack events varies. There were distinguishable patterns when comparing the difference between normal and attack events. However, further analysis needs to be performed to establish a clear distinction between them.

Figure 2d,f represent the converted time series data with different time scales with the $t_s = 1$ s, $t_s = 60$ s, and $t_s = 120$ s. We found that the original network sequences data does not provide much information associated with normal and attack events, while the converted time series data clearly shows a difference between the normal and attack events. In addition, we discovered that the time scale with $t_s = 1$ s—compared with the rest of the time scales—does not clearly separate between normal and attack events. Therefore, we excluded the $t_s = 1$ s, and used $t_s = 60$ s and $t_s = 120$ s to analyze network traffic time series.

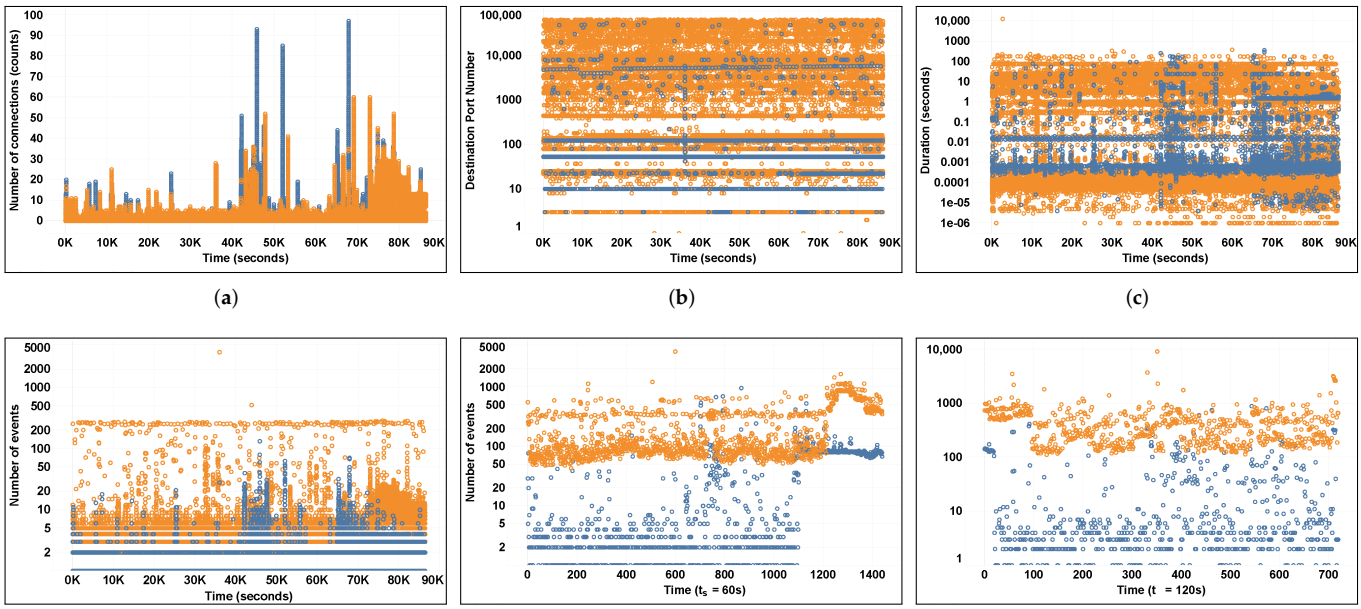


Figure 2. Representations of the network traffic data on the first day of January 2015. Orange and blue color attributes are used to indicate attack and normal events, respectively. (**a**) Total number of connections with the same source and destination IP addresses in the past two seconds, (**b**) destination port number of each network, (**c**) connection duration of each network event, (**d**) total number of events in each second, (**e**) total number of events in the converted time series data with the time scale ($t_s = 60$ s), (**f**) total number of events in the converted time series data with the time scale ($t_s = 120$ s). In (**c**,**f**), logarithmic scales are applied along the y-axis to advance the visualizations to resolve data skewness toward large-density network events.

Figure 3 displays data distributions using box plots for the data from January 2015. Due to skewed distributions in the data, logarithmic scales are applied to all figures. Figure 3a,b show box plots of the variables (i.e., source and destination bytes) associated with normal and attack events using the original data. We found that the attack events exhibited significantly higher quartiles in the destination bytes compared to normal events. But, there were only slight variations in the source bytes between normal and attack events. We also observed that the source and destination bytes in the attack events showed multiple outliers (located outside the whiskers of the box plot). As a non-parametric test, the Mann–Whitney U test is used to compare the distribution of the variables between two groups. Comparisons between the normal and attack events for the variables (source and destination bytes) were assessed using the Mann–Whitney U test. From the test, we found significant differences ($p = 0.0004$) in the distributions of the source byte variable between the normal and attack events. The destination byte variable was also determined

as significant ($p < 0.0001$) between the normal and attack events. Figure 3c,d present box plots of the total number of normal and attack events in the converted time series data with different time scales ($t_s = 60$ and 120 s), respectively.

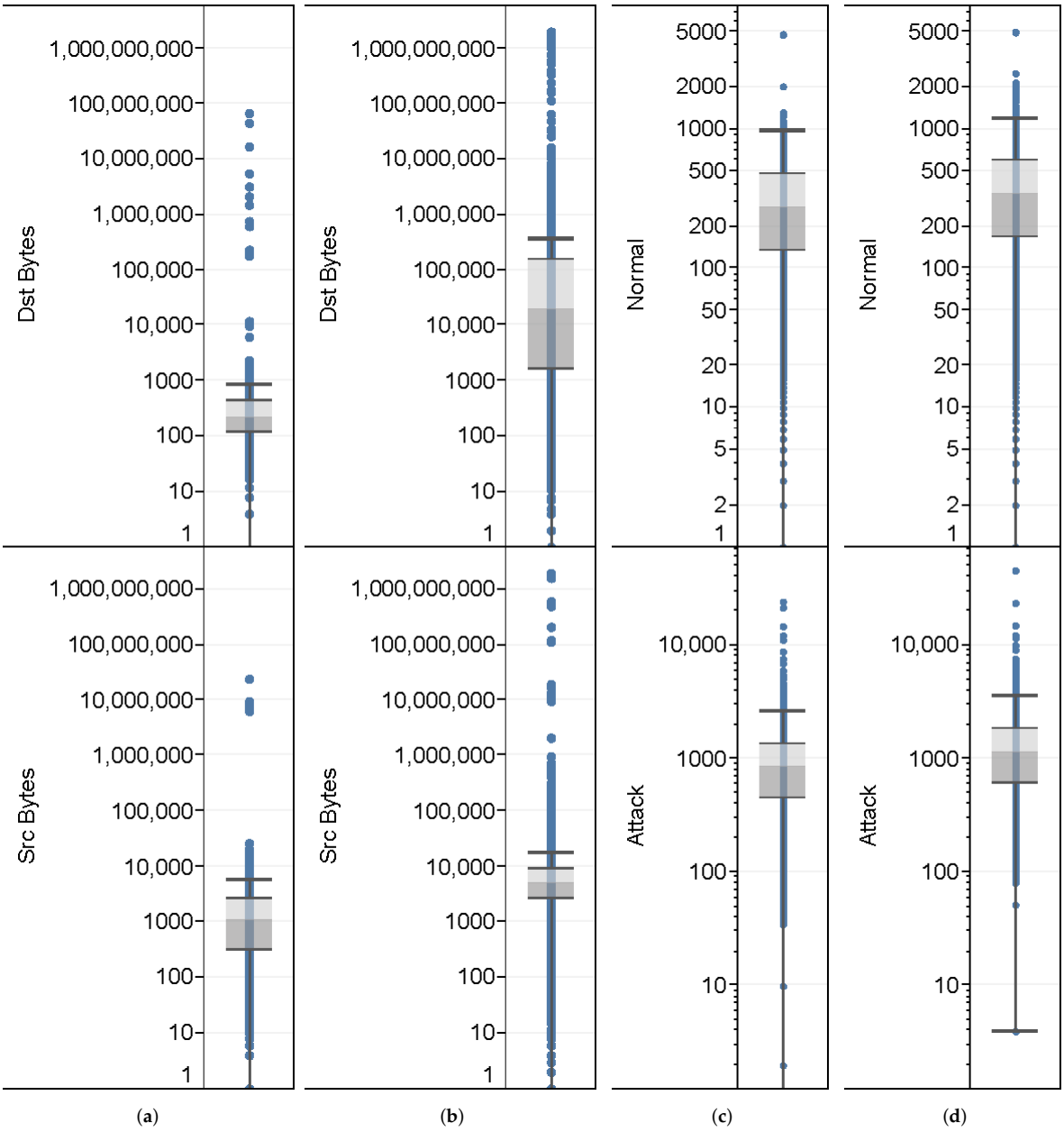


Figure 3. Box plot distributions of the network traffic data in January 2015. (**a**) Source and destination bytes of normal events, (**b**) source and destination bytes of attack events, (**c**) number of network events with the time scale ($t_S = 60$ s), and (**d**) number of network events with the time scale ($t_S = 120$ s). Logarithmic scales are applied to address data skewness.

When analyzing time series data, satisfying the stationarity property of data is essential because it influences the overall forecasting performances [39]. The augmented Dickey–Fuller (ADF) test is commonly used to check the stationarity of data. We performed the ADF test to validate if the converted time series data met the stationary assumption. The null hypothesis for the test was that the time series data were non-stationary. The test results demonstrated statistical significance ($p < 0.05$) to reject the hypothesis, indicating that the converted time series data satisfied the stationary assumption.

In time series analysis, it is important to determine the effectiveness of a particular time series in forecasting another one. The Granger causality test [40,41] is commonly utilized to validate the effectiveness of time series. Thus, we performed the Granger causality test on the converted time series data, considering both time scales ($t_s = 60$ and 120 s) and the two embedding dimensions ($e_d = 3$ and 4). This test was carried out to assess the suitability of variables to forecast a dependent variable. We tested the extracted temporal features in forecasting network events with a 95% significance. We found high similarity even with different dimensions (either $e_d = 3$ or 4). We found that 67.6% and 69% of the temporal features showed their significance with $t_s = 60$ s and 42.3% and 61.9% with $t_s = 120$ s in forecasting normal and attack events, respectively. In addition, both 53.5% and 26.8% of features were determined as significant in predicting both normal and attack events with $t_s = 60$ s and $t_s = 120$ s. We also observed that all wavelet transform and permutation entropy features ($\mathcal{W}$) demonstrated significance in predicting the attack events with $t_s = 60$ s and $t_s = 120$ s. The summary of the Granger causality test result to forecast normal and attack events is included in Table A1 in Appendix A.

The $\mathcal{P}$ features from the variables (i.e., source bytes and Dst_host_srv serror_rate) were significant to determine attack events with $t_s = 60$ s and $t_s = 120$ s. The source byte variable indicates the number of network event bytes transferred from source to destination in a single connection. The Dst_host_srv serror_rate indicates the percentage of the flag connections (i.e., s0, s1, s2, or s3) that have activated among the connections. Only the $\mathcal{P}$ features from multiple variables (duration, count, same_srv_rate, srv_serror_rate, srv_ serror_rate, dst_host_count, dst_host_same_src_port_rate, dst_host_serror_rate) were determined as significant in predicting attack events with $t_s = 60$ s and $t_s = 120$ s. Also, the $\mathcal{N}$ features from the variables (source port, destination port, TCP, and RSTOS0) were determined as significant in predicting attack events for both $t_s = 60$ and 120 s. As one of the attribute values in the flag variable, the RSTOS0 indicates the originator sends a synchronization signal (SYN) followed by a reset signal, but, an acknowledgment (ACK) of the SYN does not appear. The TCP feature is one of the attribute values in the protocol type variable. Interestingly, more nominal features were identified as significant for the $t_s = 120$ s (43.5%) to predict attack events. The nominal features from the flag variable, such as REJ, RSTO, and S0, were not determined as significant features for the $t_s = 60$ s. REJ indicates a rejected connection attempt, and S0 represents a connection attempt that has appeared but no reply. This finding suggests that nominal features might be better presented for a larger time scale. However, further analysis is required to determine an optimal time scale. We intend to address this aspect as part of our future work since finding the optional time scale is not the primary focus of this study.

Figure 4 shows the extracted nominal features (i.e., protocol types and service types) based on our proposed approach with the targeted time scales ($t_s = 60$ and 120 s). Multiple charts are generated to understand the differences between daily and monthly network events. By analyzing the variable (protocol types), we found that TCP and UDP have been commonly utilized in network communications. ICMP was also widely used in normal and attack events. This is an interesting result because ICMP is a network layer protocol for diagnosing internal network communication issues. Specifically, since the Kyoto dataset was generated in a honeypot system consisting of multiple server nodes, each node communicates continuously to check its stability using ICMP. We also observed that ICMP was used in attack events as well. However, as stated earlier, ICMP was identified as significant in predicting normal events only. Among various service types, SSH was determined as a highly applied service type in network communication. SSH supports accessing systems remotely. While SSH supports highly secure network communication, it is frequently regarded as highly susceptible, especially when not properly administered and monitored. We found numerous brute-force attacks to gain access to systems (see Figure 4d).

To analyze the daily network traffic data, we used twenty-three hours of data to generate an LSTM model. Then, the model was used to predict the normal and attack

events in the upcoming hour. For the LSTM model, the epoch and batch size numbers were set to 100 and 200, respectively. The model was generated with the learning rate (0.001) using the 'ReLU' activation function and Adam optimizer. For the loss function, MSE was evaluated. Table 1 shows forecasting performances with the targeted time scales $t_s = 60$ and 120 s in two different embedding dimensions. We found that forecasting with $t_s = 60$ s targeted time scale performed better than using $t_s = 120$ s to predict normal and attack events. Interestingly, we observed relatively high RMSE and MAE scores for the $t_s = 120$ s to forecast normal events using the March and April data. When evaluating the performances of the normal events with $t_s = 60$ s, we observed that the embedding dimension ($e_d = 4$) showed smaller RMSE values except for the months of January, March, November, and December. Furthermore, except for the months of January, March, and September, RMSE showed better performance in the attack events using the embedding dimension ($e_d = 4$). Evaluating the performances of the normal events with $t_s = 120$ s, we found that the embedding dimension ($e_d = 3$) showed smaller RMSE values except for the months of April and July. Similarly, except for the months of March, April, June, July, and November, better performance in predicting attack events was observed using the embedding dimension ($e_d = 3$).

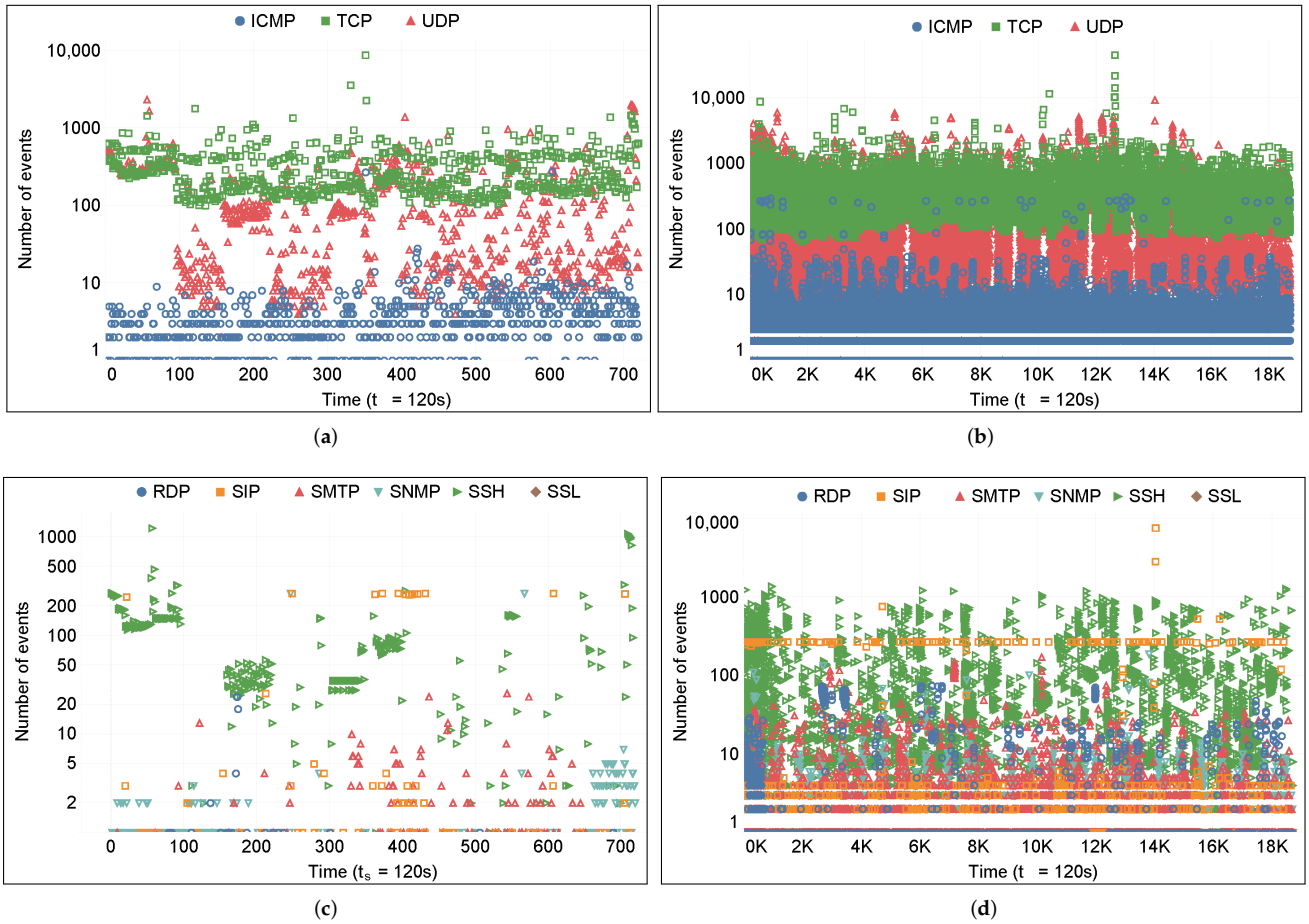


Figure 4. Representations of nominal features with the time scale ($t_s = 120$ s). (**a**) Protocol types of the network events appeared on 1 January 2015, (**b**) protocol types of all network events in January 2015, (**c**) service types of the network events appeared on 1 January 2015, and (**d**) service types of all network events in January 2015. Logarithmic scales are applied to address data skewness along the y-axis.

Figure 5 presents visual representations of predicted network events with the time scale $t_s = 60$ s by applying principal component analysis (PCA). PCA is a popular di-

mension reduction technique in visualization to represent high-dimensional data in a lower-dimensional space. The first and second principal components are determined and used to map all data instances into the x- and y-axis of the 2D space. To better represent the forecasted network events, the Fisher-Jenks algorithm (often called Jenks' natural breaks classification method) is applied to form clusters. It determines the best arrangement of values into different classes. A classification process was performed to categorize different normal and attack events into three distinct groups. Figure 5 shows grouped representations of the events with distinctive colors using 'red,' 'blue,' and 'green.' The colors represent clusters G_1, G_2, and G_3, respectively. Figure 5a,b present the clusters analyzed on 1 January 2015 (one-day data). The clusters were split into three classification groups for the normal events ($G_1 : 0 \sim 44$, $G_2 : 44 \sim 244$, $G_3 : 244 \sim$ max) and the attack events ($G_1 : 43 \sim 233$, $G_2 : 233 \sim 583$, $G_3 : 583 \sim$ max). When classification was performed with the normal events, we observed distinctively separated clusters (see G_1 : red and G_2 : blue). However, the cluster (G_3 : green) was not clearly visible. Interestingly, we identified that the cluster (G_3 : green) became distinct when analyzing the attack events (see blue-colored glyphs, Figure 5b).

Table 1. Forecasting performances using LSTM models with the standard error means for different targeted time scales (t_s) and embedding dimensions (e_d).

t_s	Month	Embedding Dimension $e_d = 3$				Embedding Dimension $e_d = 4$			
		Normal Events		Attack Events		Normal Events		Attack Events	
		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
60	January	3.96 ± 0.03	2.39 ± 0.02	5.89 ± 0.04	3.82 ± 0.03	3.27 ± 0.02	2.25 ± 0.02	4.83 ± 0.03	3.66 ± 0.03
	February	2.35 ± 0.03	1.20 ± 0.01	3.38 ± 0.02	2.30 ± 0.02	5.09 ± 0.05	2.86 ± 0.02	4.07 ± 0.02	2.95 ± 0.01
	March	7.31 ± 0.10	4.85 ± 0.07	3.87 ± 0.01	2.88 ± 0.01	7.23 ± 0.10	4.85 ± 0.07	3.74 ± 0.01	2.88 ± 0.01
	April	5.42 ± 0.11	3.79 ± 0.09	3.05 ± 0.02	2.28 ± 0.01	7.40 ± 0.08	4.90 ± 0.06	4.97 ± 0.02	3.99 ± 0.02
	May	2.66 ± 0.01	1.39 ± 0.01	4.72 ± 0.01	3.29 ± 0.01	2.67 ± 0.01	1.40 ± 0.01	4.73 ± 0.01	3.31 ± 0.01
	June	3.17 ± 0.01	1.69 ± 0.01	6.33 ± 0.02	4.47 ± 0.01	3.21 ± 0.01	1.73 ± 0.01	6.61 ± 0.02	4.62 ± 0.01
	July	4.34 ± 0.04	2.75 ± 0.02	4.61 ± 0.02	3.19 ± 0.01	5.62 ± 0.06	3.28 ± 0.03	5.72 ± 0.04	3.64 ± 0.02
	August	3.33 ± 0.02	2.37 ± 0.01	4.46 ± 0.01	3.25 ± 0.01	3.35 ± 0.02	2.37 ± 0.02	4.47 ± 0.01	3.24 ± 0.01
	September	6.22 ± 0.07	2.63 ± 0.02	4.31 ± 0.02	2.73 ± 0.01	6.22 ± 0.07	2.64 ± 0.02	3.93 ± 0.01	2.69 ± 0.01
	October	3.96 ± 0.04	1.77 ± 0.01	4.24 ± 0.02	2.48 ± 0.01	4.41 ± 0.04	1.98 ± 0.01	4.88 ± 0.03	2.79 ± 0.01
	November	5.65 ± 0.04	3.62 ± 0.02	3.88 ± 0.01	2.85 ± 0.01	5.42 ± 0.04	3.59 ± 0.02	4.35 ± 0.02	2.88 ± 0.01
	December	3.57 ± 0.02	1.71 ± 0.01	3.04 ± 0.01	1.84 ± 0.01	3.53 ± 0.02	1.71 ± 0.01	3.12 ± 0.01	1.85 ± 0.01
120	January	4.42 ± 0.02	3.11 ± 0.02	6.42 ± 0.04	4.99 ± 0.04	4.31 ± 0.02	3.08 ± 0.02	6.38 ± 0.04	4.98 ± 0.04
	February	8.16 ± 0.10	4.22 ± 0.04	5.37 ± 0.03	3.65 ± 0.01	7.77 ± 0.10	4.16 ± 0.04	4.75 ± 0.02	3.57 ± 0.01
	March	11.85 ± 0.22	8.43 ± 0.17	4.97 ± 0.01	3.95 ± 0.01	11.82 ± 0.22	8.43 ± 0.17	4.98 ± 0.01	3.97 ± 0.01
	April	10.53 ± 0.15	6.78 ± 0.17	6.65 ± 0.05	5.59 ± 0.04	10.54 ± 0.15	6.79 ± 0.11	6.66 ± 0.05	5.60 ± 0.04
	May	3.82 ± 0.02	2.10 ± 0.01	5.77 ± 0.02	4.10 ± 0.01	3.69 ± 0.02	2.04 ± 0.01	5.36 ± 0.02	3.88 ± 0.01
	June	4.83 ± 0.02	2.60 ± 0.02	6.90 ± 0.02	4.99 ± 0.02	4.69 ± 0.02	2.59 ± 0.02	7.67 ± 0.03	5.10 ± 0.02
	July	5.19 ± 0.05	3.38 ± 0.03	5.13 ± 0.03	3.78 ± 0.02	5.22 ± 0.05	3.39 ± 0.03	5.18 ± 0.02	3.79 ± 0.02
	August	5.05 ± 0.05	2.89 ± 0.02	5.28 ± 0.02	3.95 ± 0.01	5.01 ± 0.05	2.85 ± 0.02	5.27 ± 0.02	3.93 ± 0.01
	September	7.02 ± 0.07	3.11 ± 0.03	5.03 ± 0.02	3.57 ± 0.01	6.39 ± 0.07	3.01 ± 0.03	5.00 ± 0.02	3.52 ± 0.01
	October	5.72 ± 0.05	2.35 ± 0.02	5.68 ± 0.02	3.55 ± 0.01	5.36 ± 0.05	2.16 ± 0.01	5.43 ± 0.02	3.40 ± 0.01
	November	6.64 ± 0.05	4.27 ± 0.03	4.87 ± 0.02	3.87 ± 0.02	6.63 ± 0.05	4.27 ± 0.03	6.23 ± 0.05	3.98 ± 0.02
	December	3.76 ± 0.03	2.00 ± 0.01	4.49 ± 0.02	2.81 ± 0.01	3.67 ± 0.03	1.98 ± 0.01	3.98 ± 0.01	2.76 ± 0.01

For the one-month data in January 2015, the three groups were determined with the split ranges for for the normal events ($G_1 : 0 \sim 111$, $G_2 : 111 \sim 517$, $G_3 : 517 \sim$ max) and the attack events ($G_1 : 0 \sim 518$, $G_2 : 518 \sim 6967$, $G_3 : 6967 \sim$ max). With the Fisher-Jenks

algorithm, we found that most network events fall into the lower classification group (i.e., G_1). Figure 5c,d show high-density regions (red-colored) representing G_1. The blue-colored classification group (G_2) does not form separated clusters. From the analysis of data from January 2015, we found that it was not easy to see the classification group (G_3) in the PCA projections because not many network events are categorized into G_3. Presenting the data on a PCA projection space aids in understanding the similarities and differences among network events. But, we could not identify clear patterns due to high similarities among a large number of network events. Consequently, parallel coordinate visualization is applied to provide a more detailed depiction of the data.

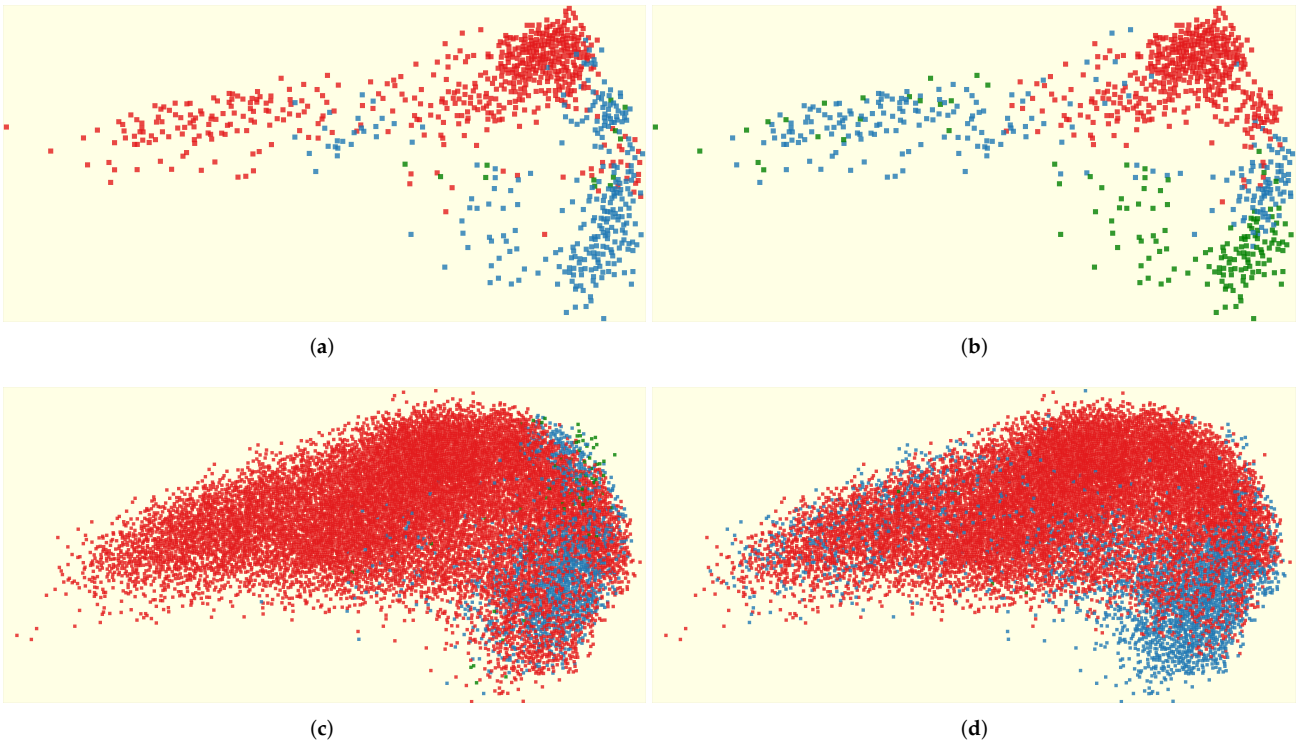
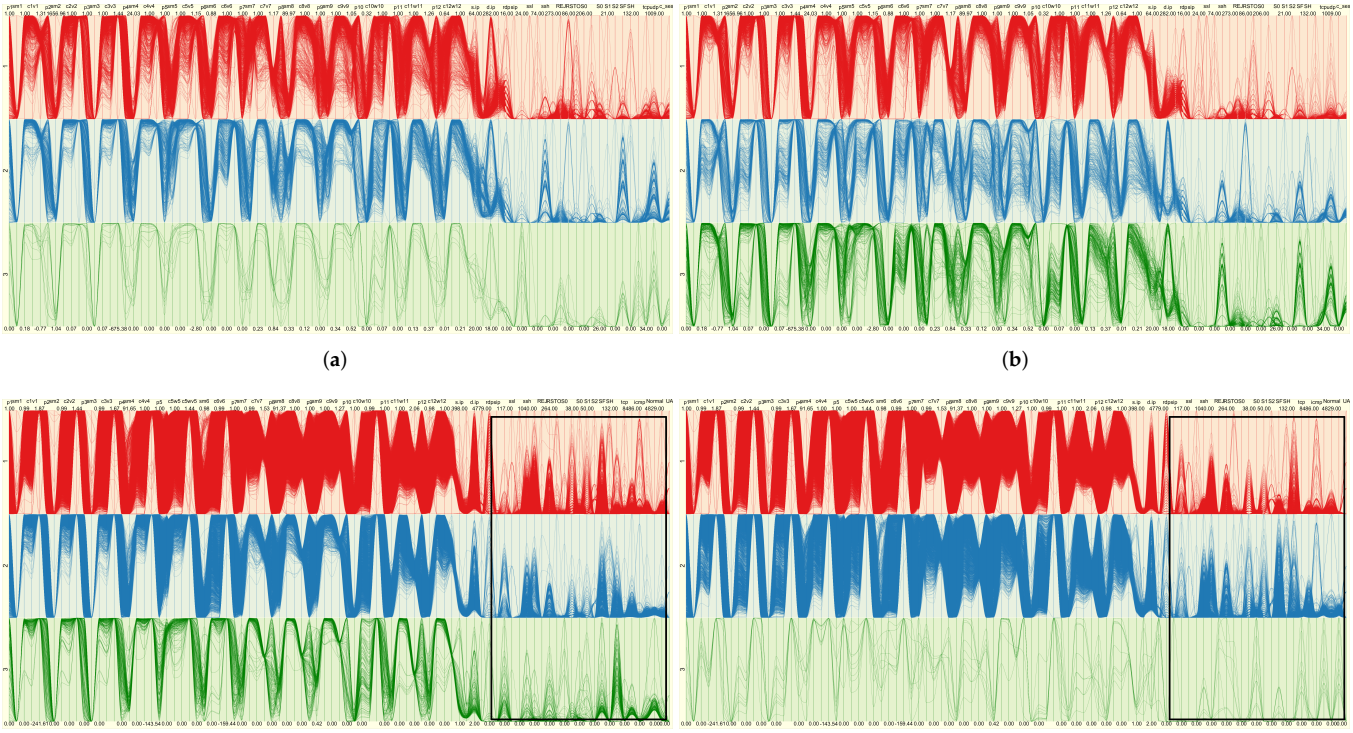


Figure 5. PCA projections by categorizing network events linearly into three groups (represented as 'red,' 'blue,' and 'green.') with determining natural breaks using the Fisher-Jenks algorithm. (**a**) PCA projection of the three groups based on the total number of projected normal events for the 1 January 2015 data, (**b**) PCA projection of the three groups based on the total number of projected attack events for the 1 January 2015 data, (**c**) PCA projection of the three groups based on the total number of projected normal events for the January 2015 data, (**d**) PCA projection of the three groups based on the total number of projected attack events for the January 2015 data.

Figure 6 shows parallel coordinates of the network events. Parallel coordinates is a visualization technique that plots individual network events as polylines in vertically arranged axes. Based on the classification group information, three separated parallel coordinates are created and arranged as the top (G_1), center (G_2), and bottom (G_3). We found high similarities between each group using the one-day data (1 January). However, when analyzing the entire January (monthly) data, we found clear differences among the groups because of the variables arranged at the end of parallel coordinates (see the black bounding box in Figure 6c,d. The variables in the bounding box represent converted nominal variables. These results suggest that nominal variables could potentially play an important role in distinguishing group patterns. However, a more comprehensive analysis is needed to understand the impact of nominal variables in distinguishing between normal and abnormal network patterns, especially across various intrusion detection datasets.



(**a**) (**b**)

(**c**) (**d**)

Figure 6. Parallel coordinates plots by categorizing network events linearly into three groups with determining natural breaks using the Fisher-Jenks Algorithm. (**a**) Parallel coordinates of the three groups based on the total number of projected normal events for the 1 January 2015 data, (**b**) parallel coordinates of the three groups based on the total number of projected attack events for the 1 January 2015 data, (**c**) parallel coordinates of the three groups based on the total number of projected normal events for the January 2015 data, (**d**) parallel coordinates of the three groups based on the total number of projected attack events for the January 2015 data. The unique distinction between each group has happened because of the variable differences in the highlighted regions (see black bounding box in (**c**,**d**)).

5. Discussion

In this study, we found that the extracted temporal features held significance in predicting network events. As demonstrated in the previous section, we found performance differences depending on the embedding dimensions ($e_d = 3$ and 4) to forecast normal and attack events. To understand the effectiveness of embedding dimensions further, we measured the statistical significance of all temporal features. 67.1%($t_s = 60$) and 58.6%($t_s = 120$) of the features were identified as significant features to forecast attack events with $e_d = 3$. With $e_d = 4$, we found similar results, as 68.6%($t_s = 60$) and 62.9%($t_s = 120$) of the features were determined significant. These results suggest that there is no significant difference between the embedding dimensions in predicting attack events. However, we found fewer significant features for predicting normal events. Specifically, 38.6%($t_s = 60$) and 37.1%($t_s = 120$) of the features were significant to forecast normal events with $e_d = 3$. On the other hand, 65.7%($t_s = 60$) and 38.6%($t_s = 120$) of the features were determined significant when using $e_d = 4$. While there was no clear difference between the dimensions in terms of the number of significant features for predicting normal events, the dimension $e_d = 4$ provided a higher number of significant features when predicting normal events.

From the analysis of forecasting performances, we found about 25% of the statistical features ($\mathcal{P}$) were significant for the time scale ($t_s = 60$ s) in forecasting both normal and attack events. However, none of the $\mathcal{P}$ features were identified as significant for the $t_s = 120$ s. We also found statistical significance in forecasting normal and attack events using the first moment features ($\mathcal{E}$). More specifically, 91.7% of the features were

significant in predicting network normal and attack events at $t_s = 60$ seconds. When using the time scale ($t_s = 120$ s), we found 75% and 66.7% of the features were determined as significant in predicting normal events, respectively. However, for predicting attack events, we found that fewer features (66.7%) were significant at predicting attack events. These results highlight the importance of utilizing the $\mathcal{E}$ features, as more than 50% of them were statistically significant for predicting both normal and attack events.

In network analysis, many previous studies have overlooked the nominal features. This study highlights the importance of using the nominal features ($\mathcal{N}$) to predict normal and attack events. When analyzing the $\mathcal{N}$ features, we found that 56.5% and 52.2% of the features were significant in predicting normal events using the time scale $t_s = 60$ and 120, respectively. However, when predicting the attack events, we found that much fewer features were significant in predicting attack events: 39.3%($t_s = 60$) and 30.4%($t_s = 120$). These results suggest that the $\mathcal{N}$ features are effective in predicting normal events (about 50% significance). But, the effectiveness of the $\mathcal{N}$ features in predicting attack events remains inconclusive, as less than 50% of the features showed significance.

By analyzing the nominal features (protocol types 'TCP,' 'UDP,' and 'ICMP'), we found 'TCP' and 'UDP' were statistically significant in predicting network events in different time scales ($t_s = 60$ and 120). But, for the 'Flag' variable representing network connection statuses ('S0,' 'S1,' 'SF,' 'REJ,' 'S2,' 'S3,' 'RSTO,' 'RSTR,' 'RSTRH,' 'SH,' 'SHR,' etc. [30]), we observed that not all features were found to be significant. In detail, the features ('RSTR,' 'RSTRH,' and 'SH') were determined as non-significant features to predict normal and attack events. The 'RSTR' variable represents a network connection that has been established but is aborted by the destination-side machine before completion. The 'RSTRH' variable indicates that the destination machine sends a 'synchronize acknowledge message' (i.e., SYN-ACK) and subsequently issues a RST (connection established) signal to abort the connection (forceful termination). SH denotes that a connection establishment message (i.e., SYN) has been sent and terminated without receiving the SYN-ACK message. Since the features ('RSTR,' 'RSTRH,' and 'SH') represent instances of incomplete network connection establishment, they were determined as non-significant variables when it came to predicting network events. In addition, the ANOVA features extracted from the duration variable were not significant.

We found that the wavelet features ($\mathcal{W}$) from DWT and PE were significant in forecasting normal and attack events. Most importantly, all wavelet features were significant in predicting attack events even with different embedding dimensions ($e_d = 3$ and 4) and time scales ($t_s = 60$ and 120). To predict normal events, we found that 91.7% of the features were significant with the time scale ($t_s = 60$). But, with the time scale ($t_s = 120$), we found about 25% of the features were significant. This may be due to the fact that data aggregation with a relatively large time scale might degrade the sudden changes. In such a case, using smaller time scales can be more effective in analyzing normal events.

As explained in Section 4, we conducted a comprehensive examination of all features using the Granger causality test to ascertain their statistical importance in predicting normal and attack events. Among the nominal variables, we found that the one-hot encoded variable ('REJ') was statistically significant in forecasting future events. Given that the 'REJ' variable represents the denial of a network connection request, it can prove valuable in predicting instances of rejected connection establishments, particularly in the context of attack events. 'TCP' and 'UDP' variables were determined as significant in predicting 70% and 69.6% of other features with $t_s = 60$ s and $t_s = 120$ s, respectively. We also found that the one-hot encoded variable 'SF' (indicating normal network connection establishment and termination) was significant in predicting 69% of the temporal variables. Interestingly, the one-hot encoded variable ('ICMP') was identified as non-significant when examining one-day and monthly data. The variables 'S0' (indicating a connection attempt was seen, but no reply) and 'SF' (referring to normal establishment and termination) were determined as significant in predicting both normal and attack events with January data (one month). But, the 'S0' variable was not significant when analyzing the January 1 data (one day)

with the Granger casualty test. The variables (source and destination port numbers) were significant in predicting network events even with different time scales. When performing the Granger casualty test, only a limited number of nominal features were recognized as significant. However, by analyzing the January data, we observed a substantial count of nominal features being designated as significant. This might be due to the necessity of substantial data volumes for accurately estimating trends in network events concerning nominal features.

We also performed the Mann–Whitney U test to compare the temporal features between $t_s = 60$ s and $t_s = 120$ s. We found that the temporal features using WT and PE showed statistical significance ($p < 0.05$). However, for the temporal features with one-hot encoded nominal variables, we could not find any statistical significance. By analyzing the January data using the Granger causality test, we found that 88.8% of the $\mathcal{P}$ and $\mathcal{E}$ features were significant for normal and attack events with the time scale ($t_s = 60$). With the time scale ($t_s = 120$ s), 50% of the $\mathcal{P}$ features and 88.8% of the $\mathcal{E}$ features were significant for normal and attack events. We also found that all $\mathcal{W}$ features were identified as significant for $t_s = 60$ s and $t_s = 120$ predicting normal and attack events. Overall, the time scale ($t_s = 60$ s) yielded better in evaluating the features and forecasting normal and attack events. By analyzing the $\mathcal{N}$ features, we found that 'S2 ($p = 0.0029$),' 'ICMP ($p = 0.0026$),' and 'SNMP ($p = 0.0005$)' were significant in predicting normal events. We observed that the variables (i.e., source and destination bytes) showed different results. In detail, the $\mathcal{P}$ and $\mathcal{W}$ features from the variables were significant in predicting network events. However, the $\mathcal{A}$ features from the variables were not significant.

We conducted an assessment of the temporal features using LSTM. Additionally, we compared two different embedding dimensions to determine an optimal dimension for analyzing the network data. While the optimal embedding dimension could not be determined clearly, the evaluation results showed different patterns depending on time scales and embedding dimensions. Specifically, with the time scale ($t_s = 60$ s) and the embedding dimension ($e_d = 4$), we found better forecasting performances to predict normal events in eight out of twelve months (66.7%) and attack events in nine out of twelve months (75%). However, when using the time scale ($t_s = 120$ s), we found that the embedding dimension ($e_d = 3$) presented better forecasting performances for normal events (88.3%—ten months out of twelve) and attack events (58.3%—seven months out of twelve). These findings suggest a potential avenue for future research to explore the relationship between time scales and embedding dimensions.

6. Conclusions and Future Work

A multivariate time series is a collection of sequences from multiple contemporaneous variables that change over time. Given the abundant information available in various application domains concerning time series data, there has been a growing interest in predicting multivariate time series. This paper presents a new approach for constructing network traffic time series at a pre-defined targeted time scale. This approach involves extracting temporal features by utilizing WT, PE, and statistical measurements to forecast normal and attack events. We also explored various techniques to extract temporal features from categorical variables by measuring the frequency of variables and creating one-hot encoded variables. The effectiveness of the proposed network activity analysis was demonstrated by comparing the performance of time series data with two targeted time scales in predicting future network event frequency using LSTM. Additionally, we employed multiple visualization techniques to analyze the time series network events, highlighting the distinctions between normal and attack events in the honeypot dataset. In future work, we plan to test different embedding dimensions to find the optimal dimension and analyze their differences across different time scales. We also plan to compare the results with different forecasting techniques, such as autoregression, vector autoregressive, and moving averages. Recently, Transformers [42] have received great attention in the time-series data analysis because they showed considerable prediction accuracy improvements over tra-

ditional methods by capturing long-range dependencies and interactions. Thus, we are going to extend our study by conducting a comparative analysis of our proposed approach with Transformers.

Author Contributions: Conceptualization, S.-Y.J. and D.H.J.; methodology, S.-Y.J., B.K.J. and D.H.J.; software, S.-Y.J. and D.H.J.; validation, S.-Y.J., D.H.J. and B.K.J.; formal analysis, S.-Y.J., D.H.J. and B.K.J.; investigation, writing—original draft preparation, S.-Y.J. and D.H.J.; writing—review and editing, S.-Y.J., B.K.J. and D.H.J.; visualization; funding acquisition, S.-Y.J. All authors have read and agreed to the published version of the manuscript.

Funding: Research was sponsored by the Army Research Office and was accomplished under Grant Number W911NF-23-1-0217. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The Kyoto dataset is available on the website at https://www.takakura.com/Kyoto_data/ (accessed on 10 September 2023). The complete analysis data and source codes will become available upon request by email.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Granger Causality Test

As described in Section 4, the Granger causality test [41] was performed on the converted time series data, considering the time scales (t_s = 60 and 120 s). The Granger causality test determines whether one time series is good for forecasting another. Table A1 shows the Granger causality test result to forecast normal and attack events. We found that about 90% of the temporal features showed their significance to forecast network events when analyzing the monthly data (i.e., January 2015 dataset).

Table A1. Evaluating the predictability of all measured variables with the Granger causality test on the data on 1 January 2015 and the entire January 2015. The variables are analyzed with the time scales (t_s = 60 and 60 s). $\mathcal{P}$, $\mathcal{W}$, $\mathcal{E}$, and $\mathcal{N}$ indicate statistical features, wavelet features, first-moment features, and nominal features, respectively. The gradient color is used to show the scale of the p-values. Statistically significant ($p < 0.05$) features are highlighted in a solid red color.

Events	**Normal**				**Attack**			
Data	**1 January 2015**		**January 2015**		**1 January 2015**		**January 2015**	
Features	**Time (t_s = 60 s)**	**Time (t_s = 120 s)**	**Time (t_s = 60 s)**	**Time (t_s)**	**Time (t_s = 60s)**	**Time (t_s = 120 s)**	**Time (t_s = 60 s)**	**Time (t_s = 120 s)**
$\mathcal{P}$	0.28	0.01	0.00	0.01	0.15	0.13	0.00	0.00
	0.01	0.23	0.00	0.00	0.00	0.01	0.00	0.00
	0.00	0.25	0.00	0.00	0.00	0.29	0.00	0.00
	0.68	0.01	0.00	0.00	0.72	0.13	0.00	0.01
	0.33	0.09	0.00	0.00	0.04	0.25	0.00	0.00
	0.23	0.29	0.00	0.00	0.01	0.12	0.00	0.59
	0.18	0.07	0.05	0.13	0.03	0.08	0.00	0.18
	0.07	0.28	0.49	0.30	0.29	0.15	0.00	0.23
	0.06	0.64	0.00	0.02	0.12	0.19	0.00	0.03
	0.01	0.34	0.00	0.00	0.00	0.13	0.00	0.00
	0.09	0.25	0.02	0.39	0.05	0.10	0.00	0.24
	0.20	0.53	0.69	0.19	0.01	0.02	0.00	0.22

Table A1. *Cont.*

Events	Normal				Attack			
Data	1 January 2015		January 2015		1 January 2015		January 2015	
Features	Time (t_s = 60 s)	Time (t_s = 120 s)	Time (t_s = 60 s)	Time (t_s)	Time (t_s = 60 s)	Time (t_s = 120 s)	Time (t_s = 60 s)	Time (t_s = 120 s)
$\mathcal{E}$	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.90	0.00	0.00	0.00	0.65	0.39	0.24
	0.80	0.63	0.42	0.31	0.62	0.66	0.79	0.76
	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.02	0.00	0.00	0.00	0.31	0.00	0.00
	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.10	0.00	0.00	0.00	0.11	0.00	0.00
	0.00	0.03	0.00	0.00	0.00	0.04	0.00	0.00
	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00
$\mathcal{W}$	0.00	0.23	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.20	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.34	0.00	0.00	0.00	0.00	0.00	0.00
	0.57	0.01	0.00	0.00	0.03	0.01	0.00	0.00
	0.00	0.47	0.00	0.00	0.00	0.01	0.00	0.00
	0.00	0.24	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.09	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.05	0.00	0.00	0.00	0.04	0.00	0.00
	0.00	0.28	0.00	0.00	0.00	0.02	0.00	0.00
	0.00	0.18	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.21	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.19	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.38	0.00	0.00	0.00	0.00	0.00	0.00
	0.43	0.00	0.00	0.00	0.02	0.00	0.00	0.00
	0.00	0.36	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.24	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.14	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.08	0.00	0.00	0.00	0.03	0.00	0.00
	0.35	0.31	0.00	0.00	0.00	0.00	0.00	0.00
	0.12	0.29	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.09	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.00	0.39	0.37	0.06	0.03	0.00	0.20
	0.50	0.02	0.01	0.72	0.36	0.82	0.00	0.57
	0.00	0.21	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.09	0.00	0.00	0.00	0.00	0.03	0.00
	0.08	0.59	0.00	0.05	0.01	0.07	0.00	0.08
	0.58	0.03	0.00	0.00	0.00	0.01	0.00	0.00
	0.00	0.09	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.12	0.00	0.00	0.00	0.00	0.90	0.00
	0.50	0.26	0.42	0.49	0.10	0.47	0.00	0.55
	0.00	0.25	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.16	0.00	0.00	0.00	0.00	0.00	0.00
$\mathcal{N}$	0.00	0.03	0.03	0.00	0.00	0.05	0.00	0.00
	0.10	0.39	0.00	0.22	0.13	0.04	0.00	0.00
	0.00	0.71	0.01	0.01	0.00	0.92	0.00	0.26
	0.64	0.42	0.58	0.43	0.67	0.26	0.05	0.00
	0.10	0.00	0.00	0.00	0.23	0.42	0.01	0.00
	0.00	0.04	0.76	0.49	0.10	0.00	0.00	0.38
	0.28	0.75	0.00	0.00	0.32	0.00	0.39	0.07
	0.00	0.80	0.00	0.00	0.46	0.06	0.08	0.00
	0.03	0.20	0.00	0.01	0.42	0.32	0.00	0.15
	0.01	0.02	0.00	0.00	0.20	0.84	0.11	0.00
	0.05	0.02	0.00	0.00	0.27	0.01	0.00	0.00
	0.47	0.06	0.00	0.00	0.00	0.43	0.00	0.00
	0.14	0.42	0.00	0.00	0.58	0.89	0.00	0.06
	0.20	0.19	0.00	0.01	0.05	0.57	0.00	0.21
	0.20	0.12	0.00	0.00	0.64	0.28	0.04	0.00
	0.00	0.81	0.02	0.00	0.07	0.00	0.00	0.07
	0.45	0.02	0.00	0.01	0.24	0.56	0.00	0.68
	0.00	0.01	0.00	0.00	0.59	0.01	0.68	0.00
	0.29	0.00	0.01	0.01	0.57	0.02	0.00	0.00
	0.00	0.00	0.00	0.00	0.04	0.43	0.00	0.00
	0.00	0.01	0.00	0.00	0.00	0.49	0.00	0.00
	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	0.00	0.05	0.00	0.00	0.37	0.94	0.00	0.80

References

1. Zhao, C.; Hu, P.; Liu, X.; Lan, X.; Zhang, H. Stock market analysis using time series relational models for stock price prediction. *Mathematics* **2023**, *11*, 1130. [CrossRef]
2. Liu, Z.; Zhu, Z.; Gao, J.; Xu, C. Forecast Methods for Time Series Data: A Survey. *IEEE Access* **2021**, *9*, 91896–91912. [CrossRef]
3. Vijay, R.K.; Nanda, S.J. Earthquake pattern analysis using subsequence time series clustering. *Pattern Anal. Appl.* **2023**, *26*, 19–37. [CrossRef]
4. Ruma, J.F.; Adnan, M.S.G.; Dewan, A.; Rahman, R.M. Particle swarm optimization based LSTM networks for water level forecasting: A case study on Bangladesh river network. *Results Eng.* **2023**, *17*, 100951. [CrossRef]
5. Yokoyama, S.; Kagawa, F.; Takamura, M.; Takagaki, K.; Kambara, K.; Mitsuyama, Y.; Shimizu, A.; Okada, G.; Okamoto, Y. Day-to-day regularity and diurnal switching of physical activity reduce depression-related behaviors: A time-series analysis of wearable device data. *BMC Public Health* **2023**, *23*, 1–9. [CrossRef]
6. Werner, G.; Yang, S.J.; McConky, K. Near real-time intrusion alert aggregation using concept-based learning. In Proceedings of the 18th ACM International Conference on Computing Frontiers, Virtual, 11–13 May 2021; pp. 152–160.
7. Montgomery, D.C.; Jennings, C.L.; Kulahci, M. *Introduction to Time Series Analysis and Forecasting*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
8. Wan, W.; Wang, Y.; Long, C.; Wei, J.; Zhao, J.; Du, G. An attack behaviors prediction model based on bag representation in time series. In Proceedings of the 2019 IEEE 44th Conference on Local Computer Networks (LCN) IEEE, Osnabrueck, Germany, 14–17 October 2019; pp. 113–116.
9. Yaacob, A.H.; Tan, I.K.; Chien, S.F.; Tan, H.K. Arima based network anomaly detection. In Proceedings of the 2010 Second International Conference on Communication Software and Networks IEEE, Singapore, 26–28 February 2010; pp. 205–209.
10. Zeng, F.; Chen, M.; Qian, C.; Wang, Y.; Zhou, Y.; Tang, W. Multivariate time series anomaly detection with adversarial transformer architecture in the Internet of Things. *Future Gener. Comput. Syst.* **2023**, *144*, 244–255. [CrossRef]
11. Abdullah, A.; Pillai, T.R.; Cai, L.Z. Intrusion detection forecasting using time series for improving cyber defence. *Int. J. Intell. Syst. Appl. Eng.* **2015**, *3*, 28–33. [CrossRef]
12. Sokol, P.; Gajdo, A. Prediction of Attacks Against Honeynet Based on Time Series Modeling. In Proceedings of the Computational Methods in Systems and Software, Szczecin, Poland, 12–14 September 2017; pp. 360–371.
13. Lee, M.C.; Lin, J.C.; Gran, E.G. RePAD: Real-time Proactive Anomaly Detection for Time Series. *arXiv* **2020**, arXiv:2001.08922.
14. Viinikka, J.; Debar, H.; Mé, L.; Lehikoinen, A.; Tarvainen, M. Processing intrusion detection alert aggregates with time series modeling. *Inf. Fusion* **2009**, *10*, 312–324. [CrossRef]
15. Fouladi, R.F.; Ermiş, O.; Anarim, E. A DDoS attack detection and defense scheme using time-series analysis for SDN. *J. Inf. Secur. Appl.* **2020**, *54*, 102587. [CrossRef]
16. Nezhad, S.M.T.; Nazari, M.; Gharavol, E.A. A novel DoS and DDoS attacks detection algorithm using ARIMA time series model and chaotic system in computer networks. *IEEE Commun. Lett.* **2016**, *20*, 700–703. [CrossRef]
17. Ergen, T.; Kozat, S.S. Unsupervised anomaly detection with LSTM neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3127–3141. [CrossRef] [PubMed]
18. Salahuddin, M.A.; Pourahmadi, V.; Alameddine, H.A.; Bari, M.F.; Boutaba, R. Chronos: Ddos attack detection using time-based autoencoder. *IEEE Trans. Netw. Serv. Manag.* **2021**, *19*, 627–641. [CrossRef]
19. Cortez, P.; Rio, M.; Rocha, M.; Sousa, P. Multi-scale Internet traffic forecasting using neural networks and time series methods. *Expert Syst.* **2012**, *29*, 143–155. [CrossRef]
20. Tian, Z. Chaotic characteristic analysis of network traffic time series at different time scales. *Chaos Solitons Fractals* **2020**, *130*, 109412. [CrossRef]
21. Mozo, A.; Ordozgoiti, B.; Gómez-Canaval, S. Forecasting short-term data center network traffic load with convolutional neural networks. *PLoS ONE* **2018**, *13*, 1–35. [CrossRef] [PubMed]
22. Yoas, D.W. Using Forecasting to Predict Long-term Resource Utilization for Web Services. Ph.D. Thesis, Nova Southeastern University, Fort Lauderdale, FL, USA, 2013.
23. Ferreira, G.O.; Ravazzi, C.; Dabbene, F.; Calafiore, G.C.; Fiore, M. Forecasting Network Traffic: A Survey and Tutorial With Open-Source Comparative Evaluation. *IEEE Access* **2023**, *11*, 6018–6044. [CrossRef]
24. Ji, S.Y.; Jeong, B.K.; Kamhoua, C.; Leslie, N.; Jeong, D.H. Forecasting network events to estimate attack risk: Integration of wavelet transform and vector auto regression with exogenous variables. *J. Netw. Comput. Appl.* **2022**, *203*, 103392. [CrossRef]
25. Wang, L.; Qu, Z.; Li, Y.; Hu, K.; Sun, J.; Xue, K.; Cui, M. Method for Extracting Patterns of Coordinated Network Attacks on Electric Power CPS Based on Temporal–Topological Correlation. *IEEE Access* **2020**, *8*, 57260–57272. [CrossRef]
26. van Heerden, R.; Malan, M.M.; Mouton, F.; Irwin, B. *Human Perception of the Measurement of a Network Attack Taxonomy in Near Real-Time*; Kimppa, K., Whitehouse, D., Kuusela, T., Phahlamohlaka, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; pp. 280–292.
27. Wang, A.; Chang, W.; Chen, S.; Mohaisen, A. Delving Into Internet DDoS Attacks by Botnets: Characterization and Analysis. *IEEE/ACM Trans. Netw.* **2018**, *26*, 2843–2855. [CrossRef]
28. Wawrowski, Ł.; Michalak, M.; Białas, A.; Kurianowicz, R.; Sikora, M.; Uchroński, M.; Kajzer, A. Detecting anomalies and attacks in network traffic monitoring with classification methods and XAI-based explainability. *Procedia Comput. Sci.* **2021**, *192*, 2259–2268. [CrossRef]

29. Taha, A.; Hadi, A.S. Anomaly Detection Methods for Categorical Data: A Review. *ACM Comput. Surv.* **2019**, *52*. [CrossRef]
30. Song, J.; Takakura, H.; Okabe, Y.; Eto, M.; Inoue, D.; Nakao, K. Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation. In Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, Salzburg, Austria, 10 April 2011; pp. 29–36.
31. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6. [CrossRef]
32. Cerda, P.; Varoquaux, G.; Kégl, B. Similarity encoding for learning with dirty categorical variables. *Mach. Learn.* **2018**, *107*, 1477–1494. [CrossRef]
33. Daubechies, I. *Ten Lectures on Wavelets*; SIAM: Philadelphia, PA, USA, 1992.
34. Bandt, C.; Pompe, B. Permutation entropy: A natural complexity measure for time series. *Phys. Rev. Lett.* **2002**, *88*, 174102. [CrossRef] [PubMed]
35. Lin, J. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory* **1991**, *37*, 145–151. [CrossRef]
36. Zhou, H.; Zhang, Y.; Yang, L.; Liu, Q.; Yan, K.; Du, Y. Short-term photovoltaic power forecasting based on long short term memory neural network and attention mechanism. *IEEE Access* **2019**, *7*, 78063–78074. [CrossRef]
37. DiPietro, R.; Hager, G.D. Deep learning: RNNs and LSTM. In *Handbook of Medical Image Computing and Computer Assisted Intervention*; Elsevier: London, UK, 2020; pp. 503–519.
38. Fu, R.; Zhang, Z.; Li, L. Using LSTM and GRU neural network methods for traffic flow prediction. In Proceedings of the 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC). IEEE, Wuhan, China, 11–13 November 2016; pp. 324–328.
39. Livieris, I.E.; Stavroyiannis, S.; Pintelas, E.; Pintelas, P. A novel validation framework to enhance deep learning models in time-series forecasting. *Neural Comput. Appl.* **2020**, *32*, 17149–17167. [CrossRef]
40. Granger, C.W. Investigating causal relations by econometric models and cross-spectral methods. *Econom. J. Econom. Soc.* **1969**, *37*, 424–438. [CrossRef]
41. Shojaie, A.; Fox, E.B. Granger Causality: A Review and Recent Advances. *Annu. Rev. Stat. Appl.* **2022**, *9*, 289–319. [CrossRef]
42. Wen, Q.; Zhou, T.; Zhang, C.; Chen, W.; Ma, Z.; Yan, J.; Sun, L. Transformers in Time Series: A Survey. In Proceedings of the International Joint Conference on Artificial Intelligence, Vienna, Austria, 23–29 July 2022.