Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Security Analytics for Real Time Forecasting of Cyberattacks

Amir Javed* | Mike Lakoju | Pete Burnap | Omer Rana

[1]School of Computer Science and Informatics, Cardiff University, 5, The Parade, Cardiff, Wales

Correspondence
*Corresponding author name Email: javeda7@cardiff.ac.uk

Summary

Protection of networked computing infrastructures (such as Internet of Things, Industrial Control Systems and Edge computing) is dependent on the continuous monitoring of interaction between such devices and network/Cloud-based hosts (especially in industry 4.0 environments). This real time monitoring enables an analyst to quantify evolving and emerging threats to such network infrastructures. A framework for identifying patterns in observed cyberthreats and the use of these patterns for forecasting the growth of an *emerging* threat to network infrastructure is proposed. This framework enables predicting the maximum threat intensity and the time period over which this maximum intensity is likely to occur. The proposed framework integrates: (i) continuous monitoring of device/ network activity, (ii) forecasting behaviour using exponentially weighted moving averages, (iii) utilising Fibonacci re-tracement for estimating the potential intensity of a cyberattack, (iv) linear regression for predicting response time for high risk thresholds and a machine learning strategy to predict potential risk over a pre-defined time window. Using this approach we can produce time intervals between the forecast and the actual attacks using real-world network activity data. Our results show an average lead time of around 1.75 hours, providing a window of opportunity to limit the impact of an attack and counter it.

KEYWORDS:
Cybersecurity, Security Risk, Malware, Web Security

1 | INTRODUCTION

In many industrial settings, the increasing adoption of Internet of Things (IoT) and Edge computing technologies significantly increases the potential attack surface within organisations (especially in industry 4.0 environments). The need for real-time sensing and control has led to increasing use of IoT in Industrial Control Systems (ICS), for both data acquisition and actuation. This ultimately creates the need for developing adaptive risk assessment methods and strategies. Risk assessment aids the identification, estimation and prioritisation of risks that could impact organisations. Cyber-risk assessment is a critical part of research in the “Chatty Factories” project[1], which explores innovative ways to securely collect actual product use data in near real-time to create new design instructions that can feed into the manufacturing process. By implication, the concept introduces IoT into an Industrial Control systems environment and requires methods and strategies for risk assessment for the use of these such IoT “products" which are able to communicate back to the factory. We propose a generic model for supporting a real-time risk prediction and monitoring system.

[1]https://www.chattyfactories.org/

This approach closely aligns with interest in “industry 4.0” focused on creating intelligent factories which enable manufacturing technologies to be transformed by the Internet of Things (IoT), cyber-physical systems and cloud computing Zhong et al and Lee *et al.*[1,2]. Within the Industry 4.0 ecosystem, manufacturing systems have the capabilities to monitor physical processes, making more effective use of “digital twins" of the physical world, and supporting timely decisions by leveraging real-time communication and cooperation with machines, sensors and humans Wang *et al.*[3]. Interest in Industry 4.0 has opened up factories to new potential attack vectors, threats and vulnerabilities. Existing literature in this area identifies the various risk mitigation strategies that have been adopted to counter the evolving threat landscape in smart factories. For instance, Aldmour *et al.*[4] looked into risk assessment methods for converged IoT and Supervisory Control and Data Acquisition (SCADA) systems. They highlight special considerations required to support the integration of IoT and SCADA systems. The increased dependence and demand for integration of IoT for connectivity and sensory capability potentially holds significant benefits for advances in SCADA systems but opens up disruptive threat surfaces to malicious individuals. The increased levels of adoption of such technologies have led to growing security concerns, hence the UK National Cyber Security Centre (NCSC) highlights the pivotal role of Operational Technology (OT) – describing this as technology that powers and drives the physical world which includes SCADA, Industrial Control Systems (ICS) and Distributed Control Systems (DCS) NCSC[5]. Many of these OT systems play major roles in Critical National Infrastructure (CNI). For example, SCADA systems are used in manufacturing facilities, steel making, gas pipelines, water distribution control systems, Power transmission, etc. Cherdantseva *et al.*[6].

In industrial applications, Internet of Things (IoT) has the capability to support process optimisation and effective factory management Bloom *et al.*[7]. Although IoT systems offer huge benefits, they are also vulnerable to an increasing number of Internet-enabled attacks and threats to data privacy. Consequently, IoT introduces a new layer of vulnerability, which is arguably more difficult to harden against attacks due to lower compute power on such devices Xi and Ling[8]. Within the industrial environment, IoT technology creates increased levels of security concerns because safety is often linked with security within the industrial setting. For instance, a security breach in a factory could destroy machines and cause harm to humans Bloom *et al.*[7]. In Industry 4.0, IoT is a natural extension of SCADA systems Hunzinger[9]. SCADA systems are instrumental to monitoring and controlling safety critical industrial equipment. The need for effective and adaptive risk assessment methods and strategies becomes invariably even more critical due to the evolving nature of threats from IoT. Risk assessment is used to identify, estimate and prioritise risks within various organisational operations resulting from the use and operation of information systems Blank and Gallagher[10]. Risk assessment is a key factor adopted within SCADA and ICS ecosystems, which forms part of the baseline for managing and mitigating threats Cheminod *et al.*and Sicari *et al.*[11,12] . Failure to conduct regular risk assessment can make a factory vulnerable to various threats which could have devastating effects to human life, the environment or to the economy Aldmour *et al.*[4]. With remote access to a factory environment, many users continue to utilise MS-RDP (Microsoft Remote Desktop Protocol) or `ssh` to connect to remote systems. As discussed in section 4, remote connectivity also introduces a number of additional vulnerabilities.

However, conventional risk assessment methods may not totally cater for the dynamic nature of IoT devices, due to the fact that the majority of the methods were created and proposed prior to the uptake of IoT Cherdantseva *et al.*[6].

A novel predictive model is proposed in this work to analyse trends in network activity and forecast the potential threat this is likely to have for a network (or connected devices). We estimate the *intensity* of emerging threats, aiming to identify the expected time for the *peak* in threat intensity, and then calculate the *lead-in-time* between the early warning of an attack and its maximum intensity for each identified threat. The work builds on existing literature by developing an implementation of the risk scoring model proposed by Awan *et al.*[13] to quantify potential emerging threats, and presents a novel forecasting framework that integrates statistical modelling for change detection and machine learning for prediction. The proposed model predicts short term threat intensity (risk for the next hour) and long term risk (time at which highest intensity for a particular risk is likely to be seen). The main contributions of this framework are: identifying change in activity trends within computer networks; forecasting the intensity of an emerging threat; predicting the lead-in-time to provide a window for a network administrator to respond and taking corrective action (e.g. deploy additional computational resources) to overcome the effect of the threat, thereby reducing system down time.

2 | RELATED WORK

A number of tools and methods to detect cyber threats have been reported in literature. These intrusion detection techniques can be broadly categorized into two main topics: Statistical Inference Models (SIM) and Machine Learning Models (ML-M)

TABLE 1 Summary of Related Works

Reference	Categorization	Approach
[15]	SIM	Forecasting using REgression (FORE)
[16]	SIM	Regression and Correlation
[17]	SIM	Correlation to Alerts
[18]	SIM	Correlation Model based on Hidden Markov Models
[19]	SIM	Clustering alerts into fuzzy events, and by fuzzy inter-event pattern mining
[20]	SIM	Three phase alert correlation framework
[21]	SIM	EWMA used on both Auto-Correlated and Correlated data
[22]	SIM	SMA and exponential weighted moving average (EWMA)
[23]	SIM	SMA, WMA, exponential smoothing, and linear regression
[24]	SIM	attack graph approach and attack probability using a Bayesian probabilistic model
[25]	SIM	ARIMA time series model
[26]	SIM	Markov model
[27]	SIM	plan recognition approach that uses dynamic Bayesian network theory
[28]	SIM	Bayesian model applied to patterns of daily activities
[29]	ML-M	Artificial Neural Networks (ANNs)
[27]	ML-M	Coefficient Correlation and Fuzzy Neural Networks
[30]	ML-M	Attacks Graphs
[31]	ML-M	Support Vector Machines (SVM)
[32]	ML-M	SVM and a modified K-mean algorithm
[33]	ML-M	Intelligent false alarm filter which selects an appropriate ML algorithm
[34]	ML-M	Fuzziness based semi-supervised learning
[35]	ML-M	ML on data generated by the bytecode stream

Subrahmanian[14]. Table 1 gives a summary of related work based on the intrusion detection technique used and the overall mitigation strategy adopted.

2.1 | Statistical models used for detecting intrusion

Applying regression models to network traffic, Park *et al.*[15] proposed a real time intrusion detection framework called *FORE* (**FO**recasting using **RE**gression). Their proposed model was based on analyzing the *randomness* of network traffic to detect intrusion. Watters *et al.*[16] used regression and correlation to efficiently conduct ethnographic studies of cyber attacks and to build profiles of attackers (Cyber Attackers Model Profile) in qualitative terms which could be used to forecast cyber attacks. Kim and Park[17] applied correlation to alerts from intrusion detection models to identify the relationship between alerts with the aim of increasing the semantic information available. Using this technique, various models were proposed to identify and detect an intrusion in the network including the detection of advanced persistent threats. A Correlation model based on Hidden Markov Models was proposed to detect intrusion in the network by Farhadi *et al.*[18]. Faraji *et al.*[19] proposed an online model for intrusion alert correlation by firstly clustering alerts into fuzzy events based on their similarity to previous events, and secondly by fuzzy inter-event pattern mining. Ahmadian and Rasoolzadegan[20] proposed a three phase alert correlation framework to detect alerts in real time. The model generates an alert, correlate alerts with the aid of causal knowledge discovery, and constructs an attack scenario using Bayesian models to predict the next wave of attack by creating prediction rules.

Techniques for intrusion detection using time series data have been applied widely within Cyber Security Park *et al.*and Fachkha *et al.*[15 23]. The motivation behind any time series forecasting model is to use sequences of events over time to project future outcomes by learning from previous activity Brockwell and Davis[36]. Moving averages have been shown to work well with time series data to smooth out fluctuation and highlight trends NIST[37]. Variants of moving averages like exponential weighted moving average have been used on both auto-correlated and correlated data to detect intrusion Nong *et al.*[21]. Simple moving average and exponential weighted moving average (EWMA) was used by Pontes *et al.*[22] to forecast attacks for intrusion detection and prevention system. Various forecasting methods like simple moving average, weighted moving average, exponential

smoothing, and linear regression have been used to predict a distributed denial of service attack's impact (intensity and size), with an error rate of less than 1 % Fachkha *et al.*[23]. Jinyu *et al.*[24] used an attack graph approach to plot vulnerabilities and attack probability for each network node using a Bayesian probabilistic model. Box *et al.*[25] used an ARIMA time series model, and Man *et al.*[26] implemented a and Markov model to predict network security situation. A plan recognition approach that uses dynamic Bayesian network theory was proposed by Feng *et al.*[27] to predict intrusion based on system calls. In another approach Bayesian model was applied to patterns of daily activities to predict if attacks will either increase or decrease Modi *et al.*[28]. There continue to be growing concerns about securing industrial systems. In a study on Unmanned Aerial Vehicle (UAVs) networks Garg *et al.*[38] opine that more effective security can be accomplished by looking at the interaction between attackers and defenders. To this end, they put forward a tree-based attack defense model for security analysis. Likewise, Jindal *et al.*[39] suggest that improving the detection of security events within a Distributed Renewable Energy Systems (DRESs) requires an adequate correlation between network and energy generation data. Consequently, they profile real network data and energy generation measurements from a local wind-turbine at Lancaster University.

2.2 | Data mining and Machine learning used for forecasting cyber threats

Data mining and machine learning have also been used to predict cyber attacks by mining network data for relationships that can be used for forecasting. Artificial Neural Networks (ANNs) have been used to develop a predictive model by successfully classifying network traffic into normal or abnormal(malicious) Modi *et al.*[29]. An intrusion prediction model was proposed by Feng *et al.*[27] that was based on coefficient correlation and fuzzy neural networks. A novel approach based on attack graphs was proposed by Li *et al.*[30] to predict intrusion in a network. A model based on Support Vector Machines (SVM) was used to predict intrusion in the network by Cheng[31]. Al-Yaseen *et al.*[32] proposed a multi-level hybrid machine learning model that uses SVM and a modified K-means algorithm to improve the classification efficiency for intrusion detection. Meng *et al.*[33] proposed an intelligent false alarm filter that adaptively selected an appropriate machine learning algorithm to filter out false alarms and to improve the performance of an intrusion detection system. Ashfaq *et al.*[34] proposed a fuzzy-logic based semi-supervised learning approach to detect intrusion in the network by utilising unlabelled samples assisted by supervised learning algorithms to improve the classifier performance. The model first extracted relevant data from the supplied dataset and then used SVM to classify data into normal or abnormal. In another approach a drive by download attack exploiting a Web browser vulnerability was detected using machine learning on data generated by the bytecode stream of web browser Jayasinghe *et al.*[35]. In summary, the research to date has been focused on detecting and predicting intrusion in a network by building statistical, probabilistic or machine learning models. However, to the best of our knowledge no work has been undertaken to predict the intensity of the attack and hence predict the lead-time for a particular threat. In this paper we introduce a novel approach that will alert a network administrator when the intensity of attack is likely to increase, forecast the level at which the maximum intensity of attack is reached and the lead time for that particular threat.

3 | ESTIMATING THREAT INTENSITY & IMPACT

We develop a framework to forecast the intensity of an emerging threat within a computer network with the aim of enabling a network administrator to implement defensive security measures to protect the network from potential damages. Detection of an emerging threat is particularly useful when a new IoT device has been added to a network. As the behaviour of such a device may not be known apriori (e.g. it may have firmware that may not be known or may be out of date), understanding the trends in data usage provides a useful guide to assess its behaviour. We use a number of software modules in the framework to analyse both short term and long term behaviour of the device, and use estimation techniques to detect the potential likelihood of behaviour indicative of an attack. Figure 1 shows the main components of our proposed approach.

At a given time instance, a log file containing monitored network traffic is provided as an input to a risk scoring function Awan[13] that calculates the risk score for a particular threat observed at time instance 't' based on equation 1.

$$Risk\lambda_t = \sigma_{i=1}^{n} Pr\alpha_{n|\lambda(t)} * Sev * W_{sev} \tag{1}$$

Once the risk score has been calculated, it is passed to the two predictive model: (i) a long term risk predictor, and (ii) a short term predictor (e.g. next hour). Both of these are described in more detail in the following sections.

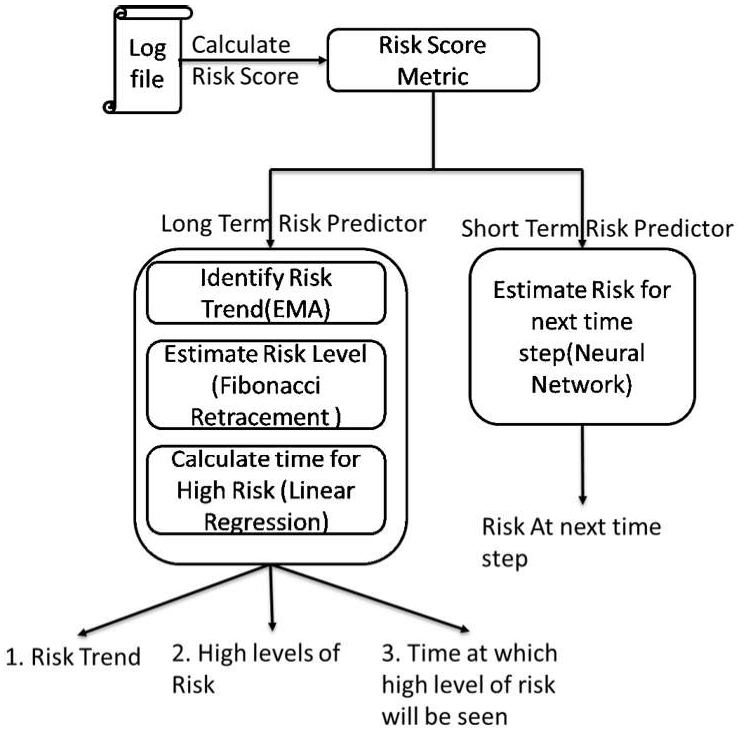


FIGURE 1 Pattern discovery & risk forecasting

3.1 | Long-Term Risk Predictor Model

In the long term risk predictor model, the risk score is calculated using equation 1 proposed by[13]. Once risk score is calculated it can be plotted over time to visualise the general trends. The long term risk predictor model is composed of three component (i) Risk Trend Identifier using an Exponential Moving Average (EMA) approach; (ii) a Risk level forecasting engine using Fibonacci retracement levels; (iii) calculation of lead in time using linear regression.

3.1.1 | Risk Trend Identifier using EMA

Once the risk score is plotted over time we use the EMA approach to identify a change in trend – identified by applying the change *"cross-over"* Neftci[40] methodology used to spot trend reversal. The EMA approach is used in order to handle the weaknesses of simple moving average by adding weights to the data points in a time series for smoothing out the effects of trend volatility and more clearly identifying the change in the observed trend NIST[37]. Using EMA the most recent data gets the highest weight, with weight assigned to older data (going back in time) decreasing exponentially NIST[41]. An EMA can be presented by the following equation 2:

$$EMA_t = \sigma Y_t + (1-\sigma) EMA_{t-1} for t = 1, 2, \ldots, n. \tag{2}$$

Where EMA_t is the targeted value, Y_t is observation at a particular time "t" and "N" is the number of observations made. σ represents the influence of the present value over the EMA outcome. For instance if σ is equal to 1 then the most recent value will have a high influence on the outcome of the EMA.

3.1.2 | Risk Level Forecasting using Fibonacci Retracement Levels

We use Fibonacci retracement levels Elliot[42] to identify levels of increasing risk by estimating two extreme points based on a defined observation period in a time series. One extreme point will represent the point where the plotted risk score is dropping while another represents the point it stops increasing. The vertical distance between these two points is divided into key Fibonacci ratios, which are calculated based on the Fibonacci number series. We have considered 23.6%,38.2%, 50%, 61.2%,100% and

161.8% Fibonacci ratios for our analysis of risk score. A Fibonacci number series can be generated by adding the last two numbers in the series, for example, 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, etc. One of the key characteristics of such a series is that each number is approximately 1.618 time greater than the previous number. The Fibonacci ratios used to identify the strategic points in the risk score chart are mathematical relationships that are expressed as ratios which are derived from Fibonacci sequences. Each ratio can be derived using the Fibonacci sequence as follows:-

$$F_{100\%} = \left(\frac{1+\sqrt{5}}{2}\right)^{0} = 1 \tag{3}$$

The ratio, 61.2%, is obtained by dividing any number in the series with the number that follows it and is described as 'the golden ratio'.

$$F_{61.8\%} = \left(\frac{1+\sqrt{5}}{2}\right)^{-1} \approx 0.6180 \tag{4}$$

The ratio, 38.2%, is obtained by dividing any number in the series with the number found at two places to the right.

$$F_{38.2\%} = \left(\frac{1+\sqrt{5}}{2}\right)^{-2} \approx 0.381966 \tag{5}$$

The ratio 23.6% is obtained by diving any number in the series with 3rd number to the right.

$$F_{23.6\%} = \left(\frac{1+\sqrt{5}}{2}\right)^{-3} \approx 0.23606 \tag{6}$$

The 50% ratio is obtained by dividing the number 1 and the number 2.

$$F_{50\%} = \left(\frac{1}{2}\right) = 0.5 \tag{7}$$

Fifth ratio, 161.8%, represents the remarkable characteristic of Fibonacci's series that each number is approximately 1.61 times greater than its previous number. for Each Fibonacci ratio that is calculated will represent the predicted risk score for a particular threat. Hence at the end of this stage we would have identified the point where the risk score has started to increase and then using Fibonacci retracement techniques for calculating the ratio for which we have predict the risk scores.

The Fibonnaci retractment approach considered in this work has been adapted from work in financial markets, taking two extreme points (usually a major peak and trough) on a stock chart and dividing the vertical distance by the key Fibonacci ratios (identified above). For reasons that are not fully clear, Fibonacci ratios play an important role in the stock market, just as they do in nature, and can be used to determine critical points that cause an asset's price to reverse. Our hypothesis in this work is that a similar Fibonnaci retractment approach can also be used to investigate growth (peaks) and falls (troughs) in the intensity of cyberattacks – and we verify this with real data acquired from an intrusion detection system in section 4. Fibonnaci retractment is based on capturing numerical anomaly within a time series and is therefore subjective – often there is no logical proof to verify this. This work is the first to utilise this approach for analysing a cyber attack time series – with each level/ratio identifying a potential area of interest to a systems administrator. A key anology we make is the concept of "value levels" used in stock markets to identify when an investor should purchase or sell a stock. Our hypothesis is that a similar approach can be used in managing cyberthreats, i.e. when determining how intense a threat has become to initiate suitable counter measures.

3.1.3 | Calculating Lead in Time

Once we have entered this stage we know the predicted risk score but we do not know the time at which this risk score will be seen. In order to calculate the risk score we have used linear regression as one of the statistical tool to calculate the time for each risk score value that is predicted using the Fibonacci retracement techniques. Linear regression is an approach which is used to model the relationship between the scalar dependent variable y (risk score for a particular threat) and that of an explanatory variable x (the time interval at which risk for a particular threat was recorded) Hellwig[43]. The output of the long term risk prediction model provides:

- The trend associated with a risk for a particular threat (rising / downward);
- Potential maximum risk level based on Fibonacci retracement;

- Time period at which these levels will be seen.

3.2 | Short Term Risk Prediction Model

TABLE 2 Performance of recurrent neural network (RNN) and Long short-term memory(LSTM)

Model	RSME on Training Dataset	RSME on Testing Dataset
RNN	1.96	1.50
LSTM	2.08	1.39

Research has shown that recurrent neural network are suitable for time series prediction Gosh *et al.*[44] – hence their use to support our short term risk prediction model. We use Long Short Term Memory (LSTM) and Recurrent Neural Networks (RNN) to identify the best model for our data to predict the risk for the next time step. Both models are built to predict the risk score using the Keras library MIT[45]. In both the models the network has a visible layer with 1 input, a hidden layer with 4 LSTM blocks or neurons, and an output layer that makes a single value prediction. The default sigmoid activation function is used for the neurons. The network is trained for 200 epochs and a batch size of 1 is used to build the model. Once both the model were built, they were trained and tested on real world (sample) data. Table 2 shows results from both the model. Results showed that a simple RNN network over performed the LSTM model during the training phase. However when LSTM out performed the simple RNN on the test data sample. Based on the performance of LSTM on the testing dataset we chosen LSTM over simple RNN to build our short term risk prediction model. The rationale for using two predictive model is that by using the long term risk prediction component we will be able to identify the point in time when the risk level starts to increase, forecast the time at which this level of potential risk will be seen. However, the short term predictive model is more of an adaptive model because a model is trained on each step incorporating the new values to make it more adaptive. The idea is that if one predictive model fails the other is there to prepare the network administrator of rising risk.

3.3 | Algorithm for Predicting Risk

In this subsection, we provide an algorithmic description for using our proposed framework to predict short term and long term risk score and to predict the response time. Algorithm 1 describes the activities undertaken in our proposed framework. A log file containing traffic instances is taken as an input to estimate likely threats. The algorithm requires selecting a specific threat represented as λ and the number of EMAs to be used for analysis. In the next step for each instances of threat recorded in the log file, a risk score for λ is calculated and then used in the two predictive models.

For the short term predictive model, at each round the data containing record of all the threats is normalised and split into two parts. Part one is used as a training dataset and part two as a validation dataset. Upon successful validation of the model, the model predicts the risk score for the next time step (T+1). Simultaneously, the risk score for the threat λ, along with number of EMA parameters to be used is passed to the long term predictive function.

Algorithm 2 describes the function required to calculate risk using the risk score formula. The function takes a log file and the threat represented by λ as input. The function will then calculate the risk score for that threat at time instance T. Depending on the threat and its severity (identified as a constant multiplier) Awan[13], the function returns the risk score for a corresponding threat.

Algorithm 3 is used to calculate the EMA outcome, which is used to reflect the dynamic environment and a changing threat landscape. It is generally observed that cyber-criminals adopt new methods and patterns based on finding new system vulnerabilities. Hence, recent attack patterns are considered to be more important than older ones. The algorithm takes σ , the risk score value and previously calculated EMA value as input and calculates the current EMA using equation 2. It then returns EMA for the present time 't'. The first instance of the EMA is taken as an average of the last 'n' values of risk score where 'n' represents the number of moving averages (for example in a 5 hourly EMA , 'n' would be 5).

Algorithm 4 describes the function that is used to predict risk score based on Fibonacci principles and it also records occurrence of each retracement level for each rising trend. For our experimental setup we have considered only five Fibonacci

Algorithm 1 Algorithm which describes the flow of control

Input: Log File (Generated by IDS containing cyber attacks occurred), Number of EMA's

Output: Output Response Time, Risk Score value for short term and long term

Select threat λ and time instance T
Select number of EMA to be applied for identifying triggers
while EOF **do**
Data=Call function Calculate Risk Score
/* Predicting Risk score for next time Step using LSTM (short term predictive model)*/
Normalise the data-set
Split the data-set into training and testing
$Train = 67\%$ of sample Data
$Test = 33\%$ of sample Data
Pre-process the data-set for LSTM function
Create and fit the LSTM network
Train the LSTM model
Use the Test data-set to test the model
Predict risk score for next time $(T + 1)$ step
Calculate root mean squared error
/* Long Term prediction for Risk score using Fibonacci */
Call Function Calculate multiple EMAs
Call Function Plot Risk Score trend-line
Call Function Plot EMAs
Identify cross over
if $(EMA_{5(t-1)} < EMA_{10(t-1)} < EMA_{15(t-1)})$ and $(EMA_{5(t)} > EMA_{10(t)} > EMA_{15(t)})$ **then** cross over= true
else cross over= false
end if
if cross over = True **then** Calculate Fibonacci retracement ratios and Response time using Linear Regression
else continue
end if
Print Response time
Print Risk Score calculate by LTPM model
Print Risk Score by STPM

Retracement levels, these are defined based on previous highest and lowest point of the wave. The five retracement levels that were chosen were 38.2%, 50%, 61.2% ,100% and 161.8%[46] of the previous highest wave. The algorithm return the predicted risk score for each threat, when the trigger signifying an increase in risk score is set off.

To summarise the short term predictive model predicts risk score based on LSTM model and the long term predictive model predicts the risk score of the threat using EMA (refer algorithm 3), Fibonacci retracement (refer algorithm 4) and linear regression techniques.

4 | DATA COLLECTION AND PREDICTIVE MEASURES

We have based our study on malicious network traffic data logs generated by the Palo Alto Networks IPS/IDS software Wildfire, which has been used as a security measure to protect more than 34000 registered users of the Cardiff University Campus network from cyber attacks. The University network has 309 building with 348 LAN rooms, 1283 switches and 3160 wireless access points. Palo Alto Networks IPS/IDS software Wildfire protects the network from both known and previously unknown malware, zero-day exploits, and Advanced Persistent Threats (APTs). Wildfire can identify over 200 potentially malicious behaviours and is capable of classifying all traffic across nearly 400 applications[47]. When an incoming traffic instance is classified as malicious,

Algorithm 2 Function to calculate Risk score , this function will take a log file as input and will calculate Risk Score for a particular threat λ.

Input: Log_t, λ
Output: $RiskScore_\lambda$

/* Initialisation of variables to calculate Risk Score*/
$W_{ser(MS-RDP)} = 15$
$Sev_{(MS-RDP)} = 0.75$
$W_{ser(APK)} = 5$
$Sev_{(APK)} = 0.25$
while end of log file **do**
  Calculate instances of threat λ in Log file
  Calculate conditional probability of λ
  Utilise values of Severity and $W_{severity}$ assigned during initialization phase
  for do$i$ = to all instance of λ
    $RiskScore_\lambda = Prob_\lambda$ at instance i $* Sev_\lambda * W_{sev} + RiskScore_\lambda$
  end for
return $RiskScore_\lambda$

Algorithm 3 This function is used to calculate the exponential weighted moving averages based on the formula: $EMA_t = \sigma Risk_\lambda t + (1-\sigma)EMA_{t-1}$ for $t = 1, 2, ..., n$. Where EMA_t is the targeted value, Y_t is an observation at a particular time 't' and 'n' is the number of observations made. The parameter σ determines the influence of present value over EMA.

Input: Log_t, λ,n,σ
Output: EMA_n

/*Function to calculate EMAσ, $Risk\_Score_n$, EMA_{n-1}, i */
Calculate σ depending on i
if if n=1 **then** $EMA_0 = Average\_Risk\_Score_i$
else
  Calculate $X = \sigma * Risk\_Score_n$
  Calculate $Y = (1-\sigma) * EMA_{n-1}$
  Calculate $EMA_n = X + Y$
end if
return EMA_n

Algorithm 4 Functions Definition This function is used to forecast risk based on Fibonacci retracement technique and Elliotts wave theory.

Input: $High_{prev\_Wave}$, Low_{Prev_Wave}
Output: Predicting Risk levels

Calculate Fibonacci re-tracement level to forecast movement based on High-Low wave range:
A=38.2% ,B=50% C=61.8% , D=100%, E=161.8%
Identify the cycle in which the wave occurs (5 wave cycle of motive and 3 wave of corrective)
while $EMA5_t > EMA10_t$ and $EMA5_t > EMA15_t$ **do**
  */*while trend is rising */*
  Monitor risk for each forecasted level
  Record data for each forecasted level seen
return forecasted risk levels

it is pushed to a RabbitMQ queuing system for storing malicious traffic logs. The data being considered in this study is typical

of many industry 4.0 environments – e.g. in the use of MS-RDP and Android SDK environments (especially for connecting to IoT systems). The attack surface observed in this data can therefore be generalised to IoT/ICS environments also.

We accessed malicious data logs of 288 hours in CSV format, obtained through RabbitMQ, for validating our proposed framework. Each malicious traffic instance had 41 attributes giving information about both source and destination IP addresses, ports, zones and countries; threats, threat categories, threat severity levels, threat occurrence time; software applications targeted; classification rule, protocols, ingress and egress interfaces as well as miscellaneous information. The malicious traffic is classified into five severity levels: 'critical', 'high','medium', 'low' and 'informational'; by the PaloAlto IDS. The log files were pre-processed to extract information about threats, threat instances, threat severity levels and threat occurrence time.

TABLE 3 Risk Score of Top 7 Threats Identified

Threat Risk Score	Min	\Max	Mean	Standard Deviation
MS-RDP	1.290	11.190	9.750	2.400
Android APK	0.000	0.130	0.010	0.015
WinDLL	0.000	0.100	0.030	0.020
WinEXE	0.000	0.160	0.030	0.060
7 Zip	0.000	1.470	0.080	0.020
PHP Query	0.000	3.790	0.230	0.660
PHP Info	0.000	1.680	0.100	0.290

The extracted information was then grouped into hourly intervals for subsequent analysis. The collected data logs of 288 hours represented two distinctive time windows each of 144 consecutive hours and contained 2 million malicious traffic instances. We then calculated the risk score using equation 1 proposed by Awan[13] for each threat represented by a malicious instance and grouped them together.

We found that 7 threats constituted 95% space of total observed threats. These seven threats that we identified were MS-RDP brute force attempt that targeted to takeover remote desktop protocol. For Windows Executable (WinExe) threat an executable file is downloaded that contained a malware. Windows Dynamic link library (DLL) that involved downloading of malicious DLL file. Android package file (APK) that targets android based phones/ devices to carry out a drive by download attack. PHP CGI Query String parameter handling information and code injection which can lead to a cyber criminal carrying out string query to extract sensitive information. Lastly, 7-Zip ARJ File Buffer overflow vulnerability that is a buffer overflow vulnerability. While all instances are important we focus mostly on MS-RDP (which comprises 92% of the instances) attacks and that of Android APK attacks (1%). The two threats were chosen particularly because we wanted to check our predictive model on threats which had the highest fluctuation or standard deviation and lowest standard deviation. We took two random sample of 144 hours each for MS-RDP and Android (APK) as inputs to our predictive model.

5 | RESULTS AND ANALYSIS

The collected data set have been used to validate the use of our proposed Risk Response Time Estimation Framework. We have determined risk scores for MS-RDP and Android Package File, and then have applied the two predictive model to the data-set. We have randomly selected 7 samples from the data set to check the effect of our predictive model. The random sample that were collected were from both Android APK and MS-RDP sample. To validate our approach, we have applied the algorithm on four random samples marked as *A* and *B* in figures 2, 4, 3, 5. Both predictive algorithms (i.e. STPM and LSTM) were applied to the samples to forecast the risk for an MS-RDP threat.

Figure 2 and 4 displays how the long term predictive model was applied to the dataset containing the risk score. One of the crucial elements of the predictive model is the EMA, which flags the change in trend, signifying if in the coming hours the risk to a network will increase or decrease. For our experiment we made use of EMA over a 4 and 8 hour window, as shown in figure 2 and 4 respectively. These two time windows were specifically chosen to identify the change in trend, by using the cross over technique. It is the point where a short term EMA (4hrs) crosses a long term EMA (8hrs). The point is known as the trigger which identifies a change in trend as pointed out by the red arrow in figure 2 and 4 – calculated using the EMA function 2.

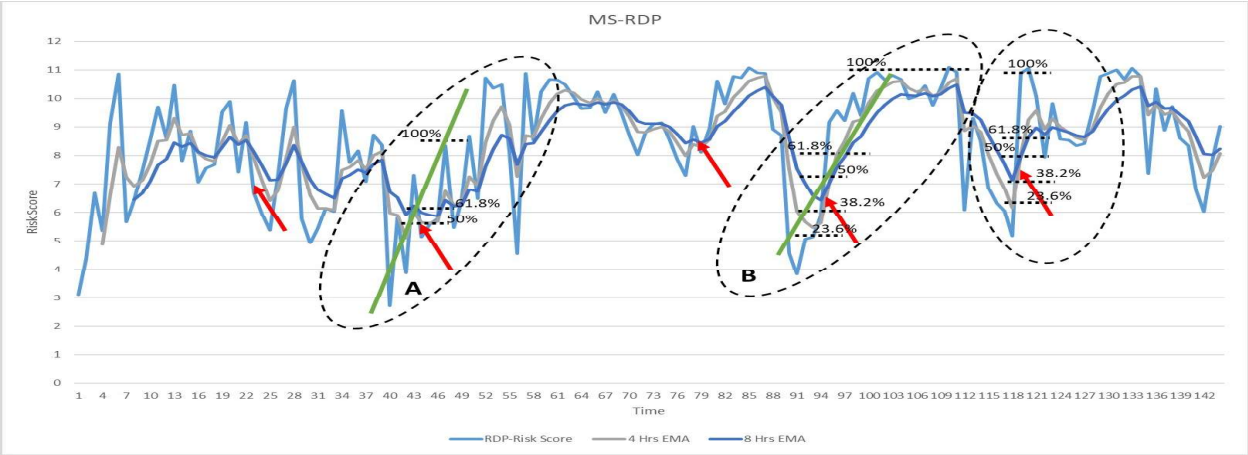
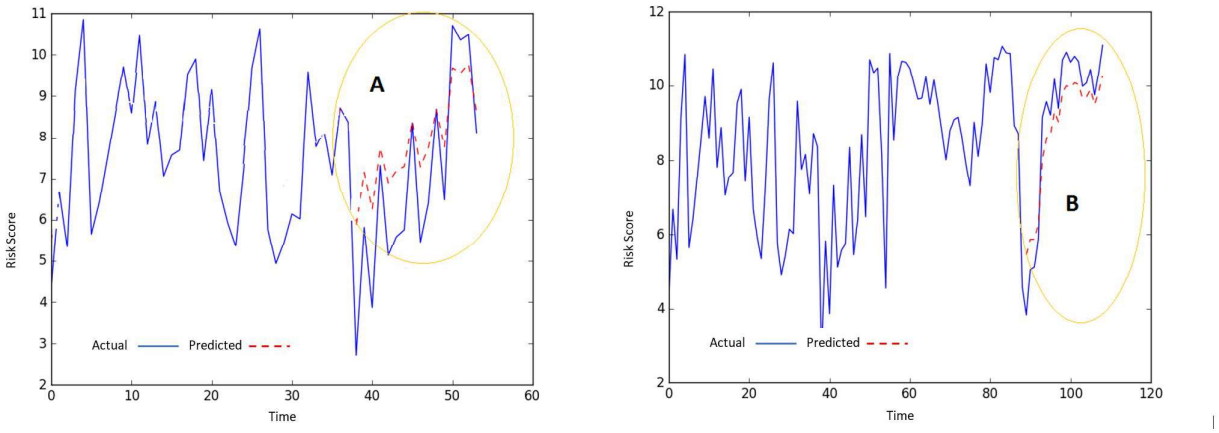


FIGURE 2 Long term forecasting model applied on MS-RDP threat's risk score chart for sample 1



(a) Short term Predictive Model applied to sample A (Shown in Figure 2)

(b) Short term Predictive Model applied to sample B (Shown in Figure 2)

FIGURE 3 Prediction of risk using both long term and short predictive model

Once a cross over trigger is detected (as shown in the area marked as *A, B* in figures 2 and 4) then Fibonacci levels are identified using a previously high-low *wave* (a wave is define as the highest and lowest point seen in the last 50 hours). If a rising trend is identified then using highest and lowest risk score values, Fibonacci levels (23.6%,38.2% ..100%) are identified as shown in figure 2 and 4. The vertical distance of the next high wave is predicted along with the possible estimated values of Fibonacci re-tracement ratios on the rising wave trend. Once we have identified the most probable level the wave can reach, based on Fibonacci ratios, the algorithm then calculates the time at which these levels will be seen using linear regression. The linear regression model (shown as the green line in figure 2 and 4) is built using the data point observed when the trend changes, i.e. the rationale is to capture the rising slope as accurately as possible. The outcome of the long term predictive model is the forecasted risk score based on Fibonacci re-tracement and the time at which these levels will be seen based on linear regression.

Figures 3 and 5 show the results of the short term predictive models using LSTM to estimate risk for next time step. In the STPM model, first the risk score is calculated (refer to algorithm 1) and then the data is divided into parts for training and validating the model. A risk score for the next time step is predicted and finally the root mean squared error is calculated for that time step. Figures 3 and 5 show the outcome of the short term predictive model. The red dotted line represents the predictive risk score and the solid blue line represents the actual risk for that time.

Similarly, both models were applied to the entire dataset to forecast risk at every time step and figure 6 gives an overview of how both models performed over the sample set. The long term predictive model performed quite well and gave an average error percentage of 3.33% whereas the short term predictive model that is based on LSTM network gave a higher error percentage of

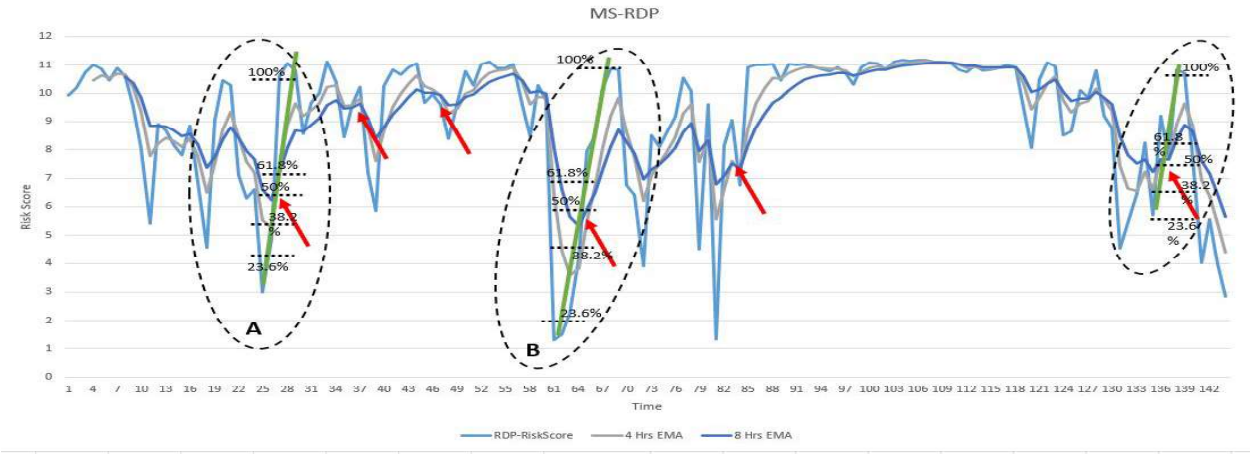
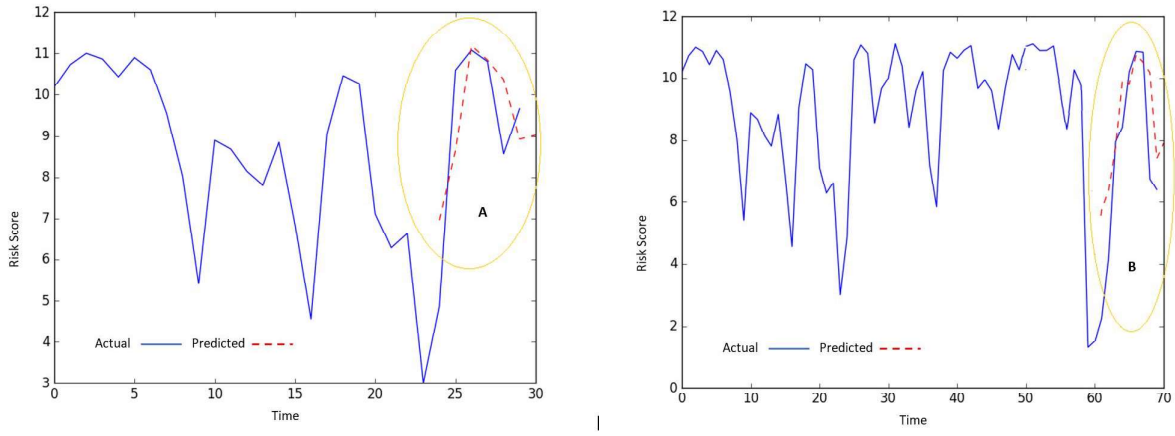


FIGURE 4 Long term forecasting model applied on MS-RDP threat – risk score chart for sample 2



(a) Short term Predictive Model applied to sample A (shown in Figure 4)

(b) Short term Predictive Model applied to sample B (shown in Figure 4)

FIGURE 5 Prediction of risk using short predictive model

7.02%. The key observation is that whenever there is a sudden jump or a sharp peak in the risk score the long term predictive model gives higher error percentage. There could be two reasons for this, one that the risk score levels were not seen in the past hence the Fibonacci re-tracement technique was unable to detect it, or although the risk score was predicted, the linear regression model failed to predict the time at which that risk score value will be seen. However, as the LSTM model in the short term predictive model gets trained every time before it makes a prediction for risk score for the next hour, it is able to adapt to the rising risk score and gives a lower error rate than the long term predictive model.

The details of predictive results on random sample set for both models are shown in table 4 – sample 1-4 where randomly taken from MS-RDP dataset whereas the rest were taken from Android APK application.

The long term predictive model was able to identify the change in trend in all the samples, meaning it triggered an alert to the network administrator that the intensity of risk for that particular threat is about to increase. This in itself is a warning signal that can indicate to the network administrator that the number of attacks are going to increase. Once the trigger is activated, the Fibonacci retracement techniques is used to forecast the risk based on previous highest and lowest values. For MS-RDP the model was able to predict the risk score with 99.02% accuracy and 94.64% accuracy for Android APK. Furthermore, by using these forecasted risk levels the model was further able to predict an average the lead in time of 3.5 hour for MS-RDP and 4.6 hours for Android APK.

For the same sample data set when we looked at the short term predictive model that used LSTM for its prediction, we achieved an accuracy of 93.56% for MS-RDP and for Android APK we achieved an accuracy of 74.56%. The accuracy of Android APK

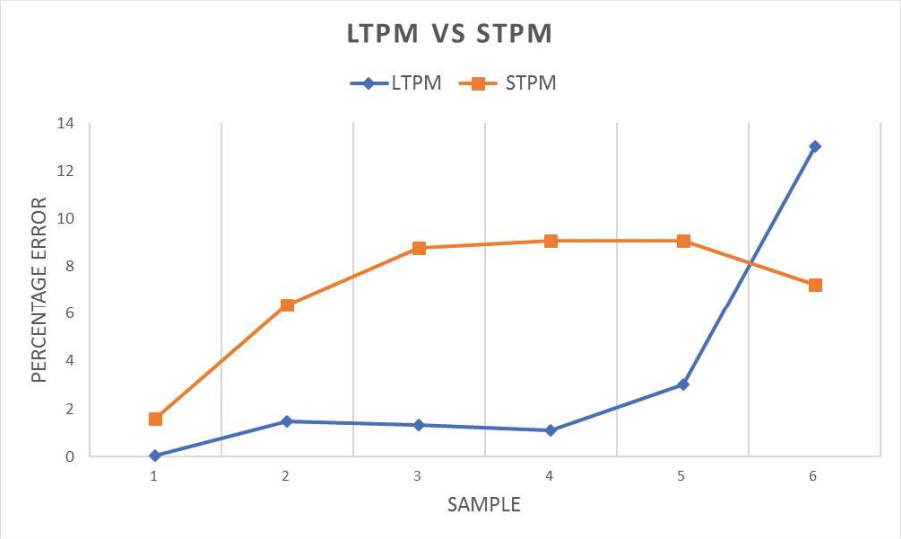


FIGURE 6 Comparison on accuracy of Long Term Predictive Model with Short Term Predictive Model

TABLE 4 Performance of the proposed prediction tool

Sample	LTPM			STPM		Actual Risk Score	Error % LTPM	Error % STPM
	EMA Trigger	Risk Forecasted	Lag Time	Risk Forecasted	Lag Time			
1	Yes	11	3	11.18	1	11	0.04	1.57
2	Yes	11.02	2	10.17	1	10.86	1.47	6.35
3	Yes	10.85	5	9.77	1	10.71	1.31	8.78
4	Yes	11.01	4	9.9	1	10.89	1.1	9.09
5	Yes	0.03	8	0.03	1	0.03	3.03	9.09
6	Yes	0.03	4	0.03	1	0.03	13.04	7.25
7	Yes	0.05	2*	0.02	1	0.05	0	60

was less because of the behaviour of the threat. The trend for Android APK varied over a narrow range band with deviation of 0.015 only. However, there were sudden peaks in attacks due to which the model gave more errors. However, since the model is adaptive it incorporates this peak and retrains itself and improves over time.

TABLE 5 Summary of predictive model

Risk	Short term Prediction (STPM)		Long Term Predictive model (LTPM)		
MS-RDP	Train RMSE	Test RMSE	Acc. of detecting trigger EMA	LR (Abs. % deviation from actual)	Average response time predicted
Sample 1	1.71	1.52	88.90%	18.40%	2.44
Sample 2	2.09	1.41	92.30%	12.30%	1.31
Android APK	Train RMSE	Test RMSE	Acc. of detecting trigger EMA	LR (Abs. % deviation from actual)	Average response time predicted
Sample 1	0.01	0.01	69.23%	29.90%	2
Sample 2	0.02	0.01	90.90%	21.90%	1.25

6 | CONCLUSION

We collected network traffic logs and identified activity surrounding two particular threats: MS-RDP and Android APK, for 288 hours. The monitoring time was divided into two intervals which were randomly selected from a spool of 144 hours of recorded data from log files. We developed a novel framework that give a short term prediction and a long term prediction. The long

term prediction is given using Exponential Moving Averages (EMA), Fibonnaci retracement and linear regression to project increasing threat levels such that network administrators can be provided with actionable insights in real-time. The short term prediction is given using LSTM predictive model that will give the network administrator a short term view of what the risk will be in the next time step. Table 2 gives a short summary of the experiment we have conducted. For the short term prediction model, the data was first split into two parts one was used to train the model and the other was used to test the model before the forecast is made for the next time step. For MS-RDP the root mean square error for the sample one is around 4.43 and that of sample two is 3.38. However the root mean square error for Android APK is less and is equal to 0.01 and 0.02. The LSTM model is trained at every time step to make it more adapted, however the model is not able to capture a sudden spike in the risk. For the Long term prediction model there are three components that we look at the trigger to capture the rising trend using the exponential moving average and on average we got an accuracy of 90.6% for MS-RDP and around 80.01% for Android APK (see table 5. Once we have identified the rising trend Fibonacci retracement levels are identified based on previous high wave. These levels are then used by the linear regression model to identify time at which these levels will be seen. For MS-RDP the linear regression model was able to predict a 100% wave retracement with an average accuracy of 15.35% and on average was able to give the network administrator a head start of 1.88 hours. For Android APK as there were not many high peaks we looked at 61.8% retracement of previous high peaks and by using the linear regression model to predict this 61.8% retracement we got an average deviation from actual data accuracy of 25.9%. For Adroid the network administrator got an early warning on an average 1.625 hours before the wave retraced. By using our model a network administrator will be able to monitor risk more closely.

7 | ACKNOWLEDGEMENT

This research was part funded by the Engineering and Physical Sciences Research Council (EPSRC) - projects EP/R021031/1 - Chatty Factories, and EP/K03345X/1 - Identifying and Modelling Victim, Business, Regulatory and Malware Behaviours in a Changing Cyberthreat Landscape.

References

1. Zhong RY, Xu X, Klotz E, Newman ST. Intelligent Manufacturing in the Context of Industry 4.0: A Review. *Engineering* 2017; 3(5): 616–630. doi: 10.1016/J.ENG.2017.05.015

2. Lee J, Bagheri B, Kao HA. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters* 2015; 3: 18–23. doi: 10.1016/J.MFGLET.2014.12.001

3. Wang S, Li D, Zhang C. Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. *Computer Networks* 2016; 101: 158–168. doi: 10.1016/J.COMNET.2015.12.017

4. Aldmour R, Burnap P, Lakoju M. Risk assessment methods for converged IoT and SCADA systems: review and recommendations. In: Institution of Engineering and Technology; 2019: 5 (6 pp.)–5 (6 pp.)

5. NCSC . Operational technologies. 2017.

6. Cherdantseva Y, Burnap P, Blyth A, et al. A review of cyber security risk assessment methods for SCADA systems. *Computers & Security* 2016; 56: 1–27. doi: 10.1016/J.COSE.2015.09.009

7. Bloom G, Alsulami B, Nwafor E, Bertolotti IC. Design patterns for the industrial Internet of Things. In: IEEE; 2018: 1–10

8. Xi W, Ling L. Research on IoT Privacy Security Risks. In: IEEE; 2016: 259–262

9. Hunzinger R. SCADA FUNDAMENTALS AND APPLICATIONS IN THE IoT. In: Hwaiyu Geng ., ed. *Internet of Things and Data Analytics Handbook, chapter 17*Hoboken: John Wiley & Sons, Inc. 2016 (pp. 283–293)

10. Blank RM, Gallagher PD. Guide for conducting risk assessments. tech. rep., 2012

11. Cheminod M, Durante L, Valenzano A. Review of Security Issues in Industrial Networks. *IEEE Transactions on Industrial Informatics* 2013; 9(1): 277–293. doi: 10.1109/TII.2012.2198666

12. Sicari S, Rizzardi A, Miorandi D, Coen-Porisini A. A risk assessment methodology for the Internet of Things. *Computer Communications* 2018; 129: 67–79. doi: 10.1016/J.COMCOM.2018.07.024

13. Continuous Monitoring and Assessment of Cyber Risks in Large Computing Infrastructures.

14. Subrahmanian V. *Handbook of computational approaches to counterterrorism*. Springer Science & Business Media . 2012.

15. Park H, Jung SOD, Lee H, In HP. Cyber weather forecasting: Forecasting unknown internet worms using randomness analysis. In: Springer. ; 2012: 376–387.

16. Watters PA, McCombie S, Layton R, Pieprzyk J. Characterising and predicting cyber attacks using the Cyber Attacker Model Profile (CAMP). 2012.

17. Kim YH, Park WH. A study on cyber threat prediction based on intrusion detection event for APT attack detection. *Multimedia tools and applications* 2014; 71(2): 685–698.

18. Farhadi H, AmirHaeri M, Khansari M. Alert correlation and prediction using data mining and HMM. *The ISC International Journal of Information Security* 2015; 3(2).

19. Faraji Daneshgar F, Abbaspour M. Extracting fuzzy attack patterns using an online fuzzy adaptive alert correlation framework. *Security and Communication Networks* 2016; 9(14): 2245–2260.

20. Ahmadian Ramaki A, Rasoolzadegan A. Causal knowledge analysis for detecting and modeling multi-step attacks. *Security and Communication Networks* 2017.

21. Nong Ye , Vilbert S, Qiang Chen . Computer intrusion detection through EWMA for autocorrelated and uncorrelated data. *IEEE Transactions on Reliability* 2003; 52(1): 75–82.

22. Pontes E, Guelfi AE, Kofuji ST, Silva AAA, Guelfi AE. Applying Multi-Correlation for Improving Forecasting in Cyber Security. *The Sixth International Conference on Digital Information Management (ICDIM)* 2011: 179–186. doi: 10.1109/ICDIM.2011.6093323

23. Fachkha C, Bou-Harb E, Debbabi M. Towards a Forecasting Model for Distributed Denial of Service Activities. *2013 IEEE 12th International Symposium on Network Computing and Applications* 2013: 110–117. doi: 10.1109/NCA.2013.13

24. Jinyu W, Lihua Y, Yunchuan G. Cyber Attacks Prediction Model Based on Bayesian Network. *Parallel and Distributed Systems (ICPADS), 2012 IEEE 18th International Conference on* 2012: 730–731. doi: 10.1109/ICPADS.2012.117

25. Box GEP, Jenkins GM, Reinsel GC, Ljung GM. *Time Series Analysis: Forecasting and Control*. Wiley. 5th ed. 2015.

26. Man D, Wang Y, Yang W, Wang W. Network Security Situation. 2009: 309–313.

27. Feng L, Guan X, Guo S, Gao Y, Liu P. Predicting the intrusion intentions by observing system call sequences. *Computers & Security* 2004; 23(3): 241–252.

28. Ishida C, Arakawa Y, Sasase I, Takemori K. Forecast techniques for predicting increase or decrease of attacks using bayesian inference. In: IEEE. ; 2005: 450–453.

29. Modi C, Patel D, Borisaniya B, Patel H, Patel A, Rajarajan M. A survey of intrusion detection techniques in cloud. *Journal of Network and Computer Applications* 2013; 36(1): 42–57.

30. Li Zt, Lei J, Wang L, Li D. A data mining approach to generating network attack graph for intrusion prediction. In: . 4. IEEE. ; 2007: 307–311.

31. Cheng-Bin L. A new intrusion prediction method based on feature extraction. In: . 1. IEEE. ; 2009: 7–10.

32. Al-Yaseen WL, Othman ZA, Nazri MZA. Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. *Expert Systems with Applications* 2017; 67: 296–303.

33. Meng Y, others . Enhancing Intrusion Detection Systems Using Intelligent False Alarm Filter: Selecting the Best Machine Learning Algorithm. In: IGI Global. 2017 (pp. 282–306).

34. Ashfaq RAR, Wang XZ, Huang JZ, Abbas H, He YL. Fuzziness based semi-supervised learning approach for intrusion detection system. *Information Sciences* 2017; 378: 484–497.

35. Jayasinghe GK, Culpepper JS, Bertok P. Efficient and effective realtime prediction of drive-by download attacks. *Journal of Network and Computer Applications* 2014; 38: 135–149.

36. Brockwell P, Davis R. *Introduction to Time Series and Forecasting* . 2002

37. NIST . NIST/SEMATECH e-Handbook of Statistical Methods. .

38. Garg S, Aujla GS, Kumar N, Batra S. Tree-Based Attack-Defense Model for Risk Assessment in Multi-UAV Networks. *IEEE Consumer Electronics Magazine* 2019; 8(6): 35–41. doi: 10.1109/MCE.2019.2941345

39. Jindal A, Marnerides AK, Scott A, Hutchison D. Identifying security challenges in renewable energy systems: A wind turbine case study. *e-Energy 2019 - Proceedings of the 10th ACM International Conference on Future Energy Systems* 2019: 370–372. doi: 10.1145/3307772.3330154

40. Neftci SN. Naive Trading Rules in Financial Markets and Wiener-Kolmogorov Prediction Theory: A Study of "Technical Analysis". 1991.

41. NIST . 6.3.2.4. EWMA Control Charts. .

42. Elliott RN. *Reconstruction of the Elliott Wave Principle*. Amer Classical Coll Pr . 1982.

43. Hellwig Z. *Linear regression and its application to economics*. Elsevier . 2014.

44. Ghosh AK, Schwartzbard A, Schatz M. Learning Program Behavior Profiles for Intrusion Detection.. In: . 51462. ; 1999: 1–13.

45. MIT . Keras Documentation. https://keras.io/; 2017.

46. Prechter RR. *The Basics of Elliott Wave Principles*. New Classics Library . 1994.

47. PALO ALTO NETWORKS . PALO ALTO NETWORKS: WildFire Datasheet. 2015.

AUTHOR BIOGRAPHY

[]Amir Javed Recently got his PhD from Cardiff University where he also received his MSc Degree in Information security and privacy, in 2015. He is currently a Lecutere at Cardiff University and has worked on a number research projects, as a research associate in Cyber Security Analytics in Cardiff. His thesis was on the fusion of machine learning with cybersecurity in which a predictive model was proposed to identify malicious URLs and social and content based factors are identified that aid in the propagation of these URLs. His research interest includes cybersecurity, data analytics, machine learning and security related to IoT devices.

[]Mike Lakoju Got his PhD from Brunel University London. He is currently a Data Science and Cyber Analytics Research Associate in the School of Computer Science and Informatics at Cardiff University. He is part of the Security, Privacy and Human Factors Research group. His current research -"Chatty Factories"- is focused on revolutionising the manufacturing industry by creating a system which allows products securely "talk" directly to the factory floor thereby allowing the possibility of harnessing product use data in real time. His focus in the research involves Operational Technology Security, Machine Learning, Data Visualisation, IoT Analytics, Information Technology Security and Security Architecture Modelling.

[]Pete Burnap Professor at Cardiff University and Social Computing research priority area lead in the School of Computer Science and Informatics Complex Systems research group. He has developed a reputation for data-driven, innovative, and inter-disciplinary research that broadly contributes to the growing field of Data Science, working closely with the Cardiff School of Social Sciences and School of Engineering. He is an applied computer scientist with a principal focus on data and computational methods to improve understanding, operations and decision making outside of academia, while contributing to the academic fields of Social Computing, Web Science and Cybersecurity.

[]Omer Rana Professor of Performance Engineering and lead the Complex Systems research group. His research interests lie in the overlap between intelligent systems and high performance distributed computing. He is particularly interested in understanding how intelligent techniques could be used to support resource management in distributed systems, and the use of these techniques in various application areas.

How to cite this article: A. Javed, M. Lakoju, P. Burnap, and O. Rana (2019), Security Analytics for Real Time Forecasting of Cyberattacks, *Special Issue on Big Data Analytics in Industry 4.0 Ecosystems Software: Practice and Experience*