





# AI Threat Detection and Response on Smart Networks

Konstantinos Dermetzis<sup>1,2</sup>  and Lazaros Iliadis<sup>2</sup> 

<sup>1</sup> School of Civil Engineering, Democritus University of Thrace, Xanthi, Greece  
kdemertz@fmenr.duth.gr

<sup>2</sup> Department of Physics, International Hellenic University, Kavala, Greece  
liliadis@civil.duth.gr

**Abstract.** The main goal of smart cities is to dynamically optimize the quality of life, through the application of information and communication technologies (ICT). The involved networks, require a continuous increase in data exchange, in order to intelligently control services and in particular, mechanisms that activate a higher degree of automation in the city. As many critical services are interconnected, the need for cyber security is increasing, in order to ensure data exchange protection, privacy, and better health and safety services for all citizens. The security and evolution of smart cities is based on the security of their smart networks which are activated by specific automation mechanisms, such as the SCADA networks and the pre-eminent automation systems. This paper presents the *AnomaTS*, an advanced *Machine Learning* system, for anomaly detection in sensors of SCADA networks, taking into account the temporal state of their mechanisms.

**Keywords:** Smart city · AI · Anomaly detection · Cyber threat · Computational intelligence · SCADA

## 1 Introduction

Attacks against SCADA [1] networks and in particular against the *Industrial Control Systems* (ICS) aim to undertake the mechanical control, the dynamic rearrangement of the centrifuge or the reprogramming in complex devices. The aim of such attacks is to speed up or slow down their operation, leading to the destruction or to the cause of permanent damage of all industrial equipment [2]. One of the most common attacks against SCADA industrial infrastructure is related to the case where the attacker, having first installed himself as *Man-In-The-Middle* in an *Ethernet ring* using the *Device-Level-Ring* protocol, carries out a *Stealthy Sensor* attack [3]. This is achieved by taking advantage of fieldbus communication in the industrial *EtherNet/IP* protocol.

Specifically, the *Fieldbus* protocol is used for distributed real-time control, allowing daisy-chain, star, ring, branch, and tree network topologies. The analog sensor control signals are coded using 4–20 mA measurements, while the I/O settings use messages that do not follow specific formats and sizes, as they are specified by the designer of the

control system [4]. Communication between sensors and control devices is performed via multicast EtherNet/IP connection over UDP. As IP Multicast is organized the data is transferred to UDP datagrams, using Class D address space network and the communication is done without ensuring the accurate transmission of the data to the information receivers. It should be specified that the opposite happens for the datagrams of the address spaces related to Classes A–C [1].

As each address in the Class D address space, represents the group of those who wish to receive the data, a host participates in the group for as long as it wishes by simply sending a JOIN Internet Group Message Protocol (IGMP) message. Due to the fact that there is no group owner, it is not necessary to be a member of the group in order to send data or to monitor the transmitted information. Obviously, it is generally very easy to install an intruder as *Man-In-The-Middle*. After establishing *Man-In-The-Middle*, the attacker launches a Stealthy Sensor attack. This attack configures the settings of sensors and actuators, in order to change the operation of specific mechanisms. However, this cannot be figured by the received measurements and it cannot be perceived by the offered displays of the overall system. More sophisticated forms of attack are applied against the sensors used in the control loops to collect measurements on SCADA infrastructure [3]. The sensors, which are active devices of the infrastructure network, are PLCs which are properly connected to each other, in order to allow remote monitoring and control of processes with high response speed.

This is the case even when the devices are distributed between different remote points. Communication (sending and receiving data) is achieved with the widely used SCADA MODBUS messaging protocol, which was originally published by *Modicon* (now Schneider Electric) for use with its programmable logic controllers (PLCs) [5]. It must be clarified that this is part of the *application layer protocol*, located at level 7 of the *OSI model*. Modbus Masters devices request information about the transfer of discrete/analog I/O and data logging from slave Modbus by performing a simple request-response format. A serious vulnerability of MODBUS lies in the inability of the protocol to recognize a forged slave/master IP address in the SCADA network. An attacker who performs a *Man-In-The-Middle* attack, can exploit this vulnerability and collect network MSU/MTU information from the returned messages, by sending queries containing invalid addresses [6].

Initially, the attacker selects network MSU/MTU information from the returned messages, and then he triggers a DoS (denial of service)/DDoS (distributed denial of service) attack, by sending request or response parameters, which contain malicious values related to the selection of the data field [7]. A very common attack scenario is related to the protocols and algorithmic ways of strategic control, which are used by control centers for smooth operation, cost minimization and security of power systems.

Power system safety is usually defined by a set of lower and upper limits for various system parameters, such as power of transmission line, and the allowed minimum/maximum operating frequency [8]. The control strategy is essentially a set of control commands which are sent to sensors and actuators, such as power generator adjustment points, error margins that have no effect on system's security, and various on/off commands. Possible removal of alerts when the system is out of range, as well as

the replacement of the cost function parameters, can create the conditions for an enemy attack on the control strategy, with completely disastrous results.

## 2 Anomaly Detection

Various anomaly detection techniques have been proposed in the literature [9], aiming to resolve severe cases of industrial equipment behavior deviation [10]. They can perform even when the nature of the attack is new and therefore unknown [11]. They are based on a tactic of comparing the current situation, with a model or more generally with a set of parameters that are considered to describe the normal operation of the system. To achieve these results, behavioral analysis related to key network parameters such as operating specifications, average power per time window is widely used.

Detection of abnormalities is related to other technical or heuristic forms of analysis, in order to identify patterns that help detect, identify and predict their occurrence, without leading to false alarms [9, 12]. The implementation of a powerful anomaly detection system requires [9, 13, 14]:

1. **Minimization of false positives:** False positives lead to reduced categorization performance and to potential loss of events in the future. In order to avoid the problems in question, it is necessary to implement a sensitive system, capable of carrying out warnings only for the most serious anomalies. Accordingly, it should be possible to draw up customized warning rules if additional sensitivity is needed.
2. **Alerting:** When an incident occurs, there should be real-time or near-real-time alerts to minimize the impact.
3. **Scaling:** Anomaly detection systems should be able to perform hundreds of checks on data flows over time, automatically scaling forecast methods to deal with increased demand events.
4. **Robustness:** When an anomaly occurs, the algorithm should not integrate these data points in order to estimate normal system behavior, but it should be able to avoid the anomaly, using large windows of historical data.
5. **Handle missing data:** Missing measurements may create a decomposing coherence structure that weakens the ability to predict. This should be adequately addressed by anomaly detection algorithms.
6. **Filtering:** Some anomalies are much more important than others, so it should be possible to filter them and take respective action.

The proposed *AnomaTS* anomaly detection system, seeks to understand the interactions between the mechanisms of intelligent networks and their automation processes, aiming to identify cyber-attacks. More specifically, the proposed approach creates a model that correlates the status of a system and its evolution over time, using modern Machine Learning (ML) techniques. Its target is to detect specialized cyber-attack patterns.

## 3 The Proposed Anomaly Detection Methodology

The proposed anomaly detection methodology is implemented as a system of iterative tasks that is applied on dataflows. Basically, there are three types of performed processes:

1. Data ingestion. They collect input source data in a buffer and they process them.
2. Anomaly detection. They receive measurement data from the buffer and apply anomaly detection methodologies.
3. The anomaly detection algorithm makes a real-time prediction, based on a trained model that has been trained in dead time.
4. It can maintain abnormal points and predictions in the buffer and it can display them centrally, through a centralized anomaly control panel.
5. Alerting. It takes the abnormal points from the buffer and it filters them with the configured rules, which are synthesized as concentrations of diametric measurements and they are compared to predefined boundaries
6. If these rules are followed, notifications are sent and actions are imposed.

The system is assisted by a database that enhances the anomaly detection workflow by storing the following:

1. Metrics metadata: They include measurement aliases, measurement levels and measurement relationships
2. Ingestion configuration: It determines data retention windows, data source types and endpoints.
3. Anomaly detection rules: They are defining anomaly retention time windows, model references and limits.
4. Configurations: They are related to configurations of notification rules, and anomaly visualization.

The actual anomaly detection problem, can be considered as a problem of analysis – prediction of time series [6, 13]. The aim is to find the mathematical relation that can model historical data in relation to time. The general modeling method, uses non-parametric techniques offering significant advantages over conventional methods. It gives an opportunity to overcome the statistical problems associated with the normality and linearity assumptions that are necessary in conventional or linear regression methods. The hypothesis of the underlying technique [15], suggests that the predictors have a cumulative structure, which allows their easy interpretation and modeling. At the same time, a detailed search of the transformation of each variable is not required. More specifically, the estimation of the dependent variable  $Y$  in this case, for a single independent variable  $X$ , can be given by the following Eq. 1:

$$\Upsilon = s(X) + error \quad (1)$$

Where  $s(X)$  is an unspecified smoothing function, whereas  $error$  is the error which usually has zero mean value and constant dispersion. The smoothing function can be determined, for example, by the *current mean* or by the *current median*, or by the local *least-squares*, the *Kernel*, the *Loess* or the *spline* method. The term “*current*” means the serial calculation of a statistic, which is applied to overlapping intervals of values of the independent variable, such as *running mean*. In modeling, the *classical linear hypothesis* is extended to include any error probability distribution (*Poisson*, *Gamma*, *Gaussian*, *Binomial* και *Inverse Gaussian*).

### 3.1 Description of the Dataset

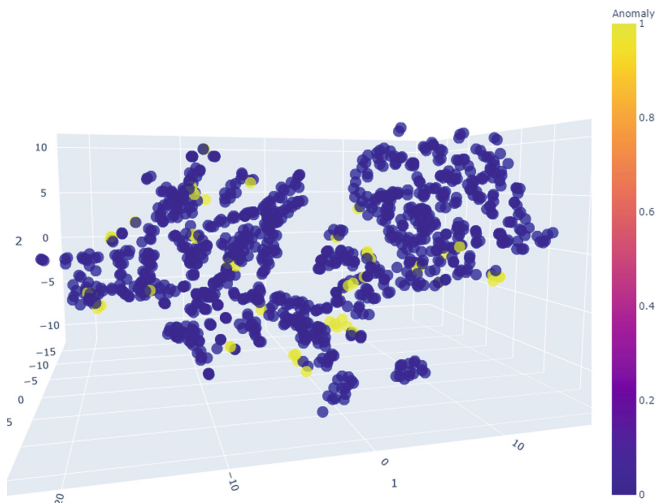
The *Factory.io* data collection platform, combined with the *InfluxDB* were employed to create an ideal simulation scenario [16]. The aim was the collection of industrial environment data, based on the *OPC-UA* open-source collector protocol. It is a collection of sensor data in the form of time series.

Sensor data is collected from construction equipment, via programmable PLC controllers and SCADA systems in order to be stored in *InfluxDB*. The storage database is optimized for timestamp or time series. Time series data is obtained by measurements or by events that are monitored and collected over time. Such events can be server metrics, application performance monitoring and transactions. Potential sources can be sensors, or various types of analytics.

In this case, one-year data were collected from hourly measurable values of three sensors, in the context of a machine condition that operates 24/7. The attack configures the sensor settings, in order to change the operation of specific mechanisms, but this is not perceived by the meters and displays of the overall system.

Specifically, there is a storage tank for raw water. This includes a water level sensor, a valve that opens when the sensor shows a level lower or equal to 0.5 m and closes when the level is higher than 0.8 m. It also contains a pump, whose action depends on a process according to which the pressure levels lead in separation through a semipermeable membrane. If the water level in the tank is below 0.25 m, the pump is immediately switched off, which is interpreted as a safety mechanism. The attacker's goal is to exaggerate the water without being detected by a standard detection mechanism based on the detection of anomalies. This is achieved by modifying the sensor and actuator information, by constructing appropriate packets, which are adapted so that the fieldbus communication can change the functionality of the devices.

A graphical representation of the anomalies contained in the time series under consideration is shown in Figs. 1 and 2 below.



**Fig. 1.** 3D plot of the time-series anomalies in the IoT (internet of things) dataset

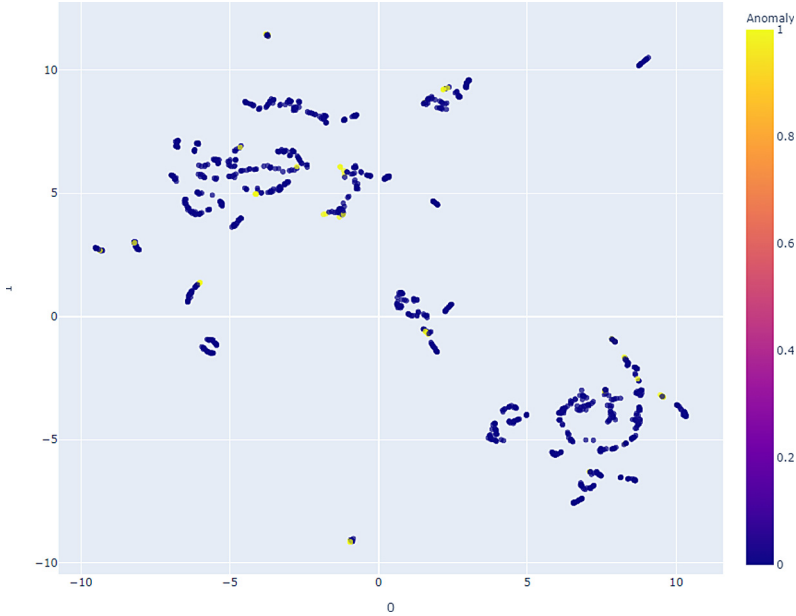


Fig. 2. 2D plot of the time-series anomalies in the IoT dataset

### 4 The Proposed Intelligent AnomaTS Algorithm

Considering the enormous difficulty of the attack scenario, the algorithm additionally receives the pressure drop measurements in a water filter present in the tank. The difference in the measurements of water pressure collected at the input from the ones collected at the output, in combination with the indications of other sensors, can give a clear sign of an anomaly related to a cyber-attack.

To solve the given scenario, the intelligent *AnomaTS* algorithm is proposed, which adapts as components, many linear and non-linear time functions, where in their simplest form three basic elements are used: trend, seasonality, and events, which are combined in the following Eq. 2 [17]:

$$y(t) = g(t) + s(t) + ev(t) + e(t) \tag{2}$$

where:

- $g(t)$ , trend models *non-periodic* changes (i.e. growth over time)
- $s(t)$ , seasonality presents *periodic* changes (i.e. weekly, monthly, yearly)
- $ev(t)$ , ties in effects of events (on potentially irregular schedules  $\geq 1$  day(s))
- $e(t)$ , covers idiosyncratic changes not accommodated by the model

A more general form of the above equation can be as follows:

$$y(t) = \textit{piecewise\_trend}(t) + \textit{seasonality}(t) + \textit{events\_effects}(t) + \textit{noise}(t)$$

In a more thorough analysis, the test variables can be deconstructed as follows:

1. *Trend*. The process includes two possible trend models for  $g(t)$ , namely a Saturating Growth Model and a Piecewise Linear Model as follows:

a. *Saturating Growth Model*. If the data suggests promise of saturation:

$$g(t) = \frac{C}{1 + \exp(-k(t - m))} \tag{3}$$

where  $C$  is the carrying capacity,  $k$  is the growth rate and  $m$  is an offset parameter.

The integration of trend changes in the model is explicitly defined by the  $S$  change points  $S_j, j = 1, \dots, S$  where the change in growth rate is located. This defines a change of settings vector  $\delta_j$  respective to time  $s_j$  with  $\delta \in R^S$ . For every moment  $t$  the rate  $k$  can be expressed as  $k + \sum_{j:t>S_j} \delta_j$ . If in this relation we estimate the vector  $\alpha(t) \in \{0, 1\}^S$  such as:

$$a_j(t) = \begin{cases} 1, & \text{if } t \geq S_j \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

The rhythm at the moment  $t$  is  $k + a(t)^T \delta$ . When the rhythm  $k$  is set, the offset parameter  $m$  must also be adapted to connect the endpoints of the sections. The correct setting at the point of change  $j$  is estimated as follows:

$$y_j = \left( S_j - m - \sum_{i<j} y_i \right) \left( 1 - \frac{k + \sum_{i<j} \delta_i}{k + \sum_{i\leq j} \delta_i} \right) \tag{5}$$

The final function is completed as follows:

$$g(t) = \frac{C(t)}{1 + \exp(-(k + a(t)^T \delta)(t - (m + a(t)^T y)))} \tag{6}$$

b. *Linear Trend with Changepoints*. It is a Piecewise Linear Model with stable development rate, estimated as follows:

$$g(t) = \left( k + a(t)^T \delta \right) t + \left( m + a(t)^T y \right) \tag{7}$$

where  $k$  is the growth rate,  $\delta$  has the rate adjustments,  $m$  is the offset parameter and, to make the function continuous,  $y_j$  is set to  $-S_j \delta_j$ .

c. *Automatic Changepoint Selection* is used to estimate the changepoints as follows:

$$\delta_j \sim \text{Laplace}(0, \tau) \tag{8}$$

where  $\tau$  directly controls the flexibility of the model in altering its rate. It should be noted that a sparse earlier adjustment  $\delta$  has no effect on the primary growth rate  $k$ , such as  $\tau$  evolves to 0 and the adjustment reduces the typical (no piecewise) logistic or linear growth.

- d. *Trend Forecast Uncertainty.* When the model deviates beyond background to make a prediction, the trend  $g(t)$  will have a stable rythm. Uncertainty in the forecast trend is estimated by extending the production model forward, where there are  $S$  change points over a history of points  $T$ , each of which has a change of pace  $\delta_j \sim Laplace(0, \tau)$ . The simulation of future rhythm changes (imitating those of the past) is achieved by replacing  $\tau$  t with a variance derived from the data. This is achieved by estimating the maximum probability of the rate scale parameter as follows:

$$\lambda = \frac{1}{S} \sum_{j=1}^S |\delta_j| \tag{9}$$

Future sample change points are randomized in such a way that the average frequency of the change points matches the corresponding historical points as follows:

$$\forall_j > T, \begin{cases} \delta_j = 0 \text{ w.p. } \frac{T-S}{T} \\ \delta_j \sim Laplace(0, \lambda) \text{ w.p. } \frac{S}{T} \end{cases} \tag{10}$$

- 2. *Seasonality.* Seasonal variable  $s(t)$  offers adaptivity to the model, allowing changes based on everyday, weekly and annual seasonality. Approximate smooth seasonal snapshots are connected to a standard *Fourier series* in order to produce a flexible model of periodic modeling.

$$s(t) = \sum_{n=1}^N \left( a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right) \tag{11}$$

- 3. *Events.* The  $ev(t)$  element reflects predictable events, including those on irregular schedules, which may create serious bias in the model. Assuming that the results of the events are independent, seasonality is calculated by the model creating a regression matrix:

$$Z(t) = [1(t \in D_1), \dots, 1(t \in D_L)] \tag{12}$$

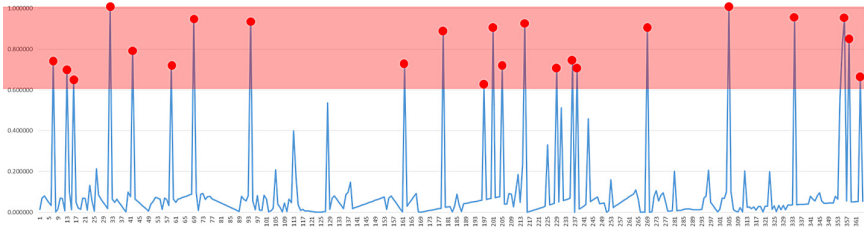
$$h(t) = Z(t)k \tag{13}$$

## 5 Running and Testing the AnomaTS Algorithm

Utilizing the procedure described above, the model was trained to detect the abnormalities that occur during the operation of the SCADA automations that control the water tank of the scenario under consideration. The class separation threshold, plays the most important and critical factor in the success or failure of the anomaly recognition method. To determine an optimal threshold, this paper proposes a reliable heuristic method of selection, based solely on evaluation criteria. In particular, the proposed algorithm assumes that a distance function is defined in the training phase, which measures the distance  $d$  between the objects and the respective target category. The threshold

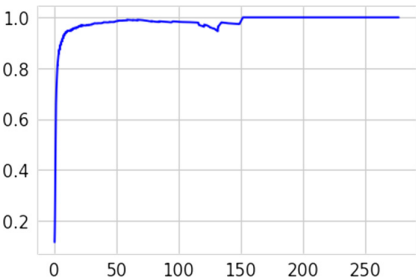


$\theta$ , is used for the binary class separation (*normal* or *abnormal*) [11, 18]. The samples (outliers) for which the anomaly score deviates from the normal operation by more than 25% are characterized as *abnormals*. This percentage emerged after a thorough analysis following a trial and error approach. Finally, the threshold  $\theta$  was set at *Anomaly score* > 0.6 in order to strengthen the classifier and isolate any divergent actions (Fig. 3).

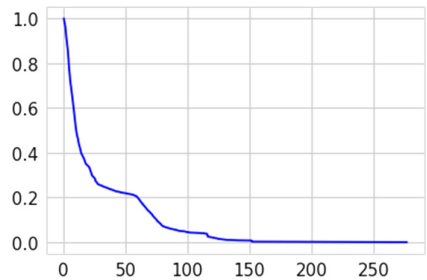


**Fig. 3.** Outliers with *Anomaly score* > 0.6

The following Figs. 4 and 5 show the results of the tests performed to select the proper threshold, that could offer the best performance.



**Fig. 4.** Precision for different threshold values



**Fig. 5.** Recall for different threshold values

The following Table 1, is the Confusion Matrix of the Binary Classification performed following the proposed *AnomaTS* method. Table 2 presents the values of the classification accuracy for five different Machine Learning algorithms.

**Table 1.** Confusion matrix

	Normal	Abnormal	
TP	80	7	FN
FP	5	273	TN

In conclusion, based on the obtained values of the performance indices and taking into account the objective difficulties raised in this research, the proposed model has been proven very efficient, able to cope with complex situations and to recognize anomalies.

**Table 2.** Classification accuracy and performance metrics

Classifier	Accuracy	RMSE	Precision	Recall	F-score	AUC
AnomaTS	96.72%	0.0841	0.967	0.967	0.967	0.9823
One class SVM	94.18%	0.0942	0.942	0.942	0.942	0.9790
Isolation forest	93.57%	0.0936	0.935	0.935	0.936	0.9712
k-NN	92.29%	0.1009	0.991	0.930	0.930	0.9697
Clustering	88.57%	0.1128	0.886	0.886	0.886	0.9464

## 6 Conclusions

An extremely innovative, reliable, low-demand and highly efficient anomaly recognition system, based on advanced computational intelligence methods, was presented in this paper. The proposed framework, utilizes advanced techniques in order to detect malfunctions or deviations from the normal operation mode of industrial equipment, which in most of the cases is due to cyber-attacks. The proposed digital security system was tested on a complex data set, which responds to specialized operating scenarios of normal and malicious behavior.

Proposals for the development and future improvements of this system, should focus on the automated optimization of the appropriate pre-training parameters, so as to achieve an even more efficient, accurate and faster classification process. It would also be important to study the expansion of this system by implementing more complex architectures with the implementation of multidimensional chronological data. Finally, an additional element that could be studied in the direction of future expansion, is the development and application of self-improvement techniques, capable of redefining its parameters automatically, so that it can fully automate the process of anomalies detection.

## References

1. Ghosh, S., Sampalli, S.: A survey of security in SCADA networks: current issues and future challenges. *IEEE Access* 7, 135812–135831 (2019). <https://doi.org/10.1109/ACCESS.2019.2926441>
2. Irmak, E., Erkek, İ.: An overview of cyber-attack vectors on SCADA systems. In: 2018 6th International Symposium on Digital Forensic and Security (ISDFS), pp. 1–5 (March 2018). <https://doi.org/10.1109/ISDFS.2018.8355379>
3. Irmak, E., Erkek, İ.: An overview of cyber-attack vectors on SCADA systems. In: 2018 6th International Symposium on Digital Forensic and Security (ISDFS), Antalya, pp. 1–5 (March 2018). <https://doi.org/10.1109/ISDFS.2018.8355379>
4. Kang, D., Kim, B., Na, J.: Cyber threats and defence approaches in SCADA systems. In: 16th International Conference on Advanced Communication Technology, pp. 324–327 (Feb. 2014). <https://doi.org/10.1109/ICACT.2014.6778974>

5. Deng, L., Peng, Y., Liu, C., Xin, X., Xie, Y.: Intrusion detection method based on support vector machine access of Modbus TCP protocol. In: 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCoM) and IEEE Smart Data (SmartData), pp. 380–383 (Dec. 2016). <https://doi.org/10.1109/iThings-GreenCom-CPSCoM-SmartData.2016.90>
6. Aminuddin, M.A.I.M., Zaaba, Z.F., Samsudin, A., Juma'at, N.B.A., Sukardi, S.: Analysis of the paradigm on tor attack studies. In: 2020 8th International Conference on Information Technology and Multimedia (ICIMU), pp. 126–131 (Aug. 2020). <https://doi.org/10.1109/ICIMU49871.2020.9243607>
7. Al-Hawawreh, M., Sitnikova, E.: Leveraging deep learning models for ransomware detection in the industrial internet of things environment. In: 2019 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, pp. 1–6 (Nov. 2019). <https://doi.org/10.1109/MilCIS.2019.8930732>
8. Al-Hawawreh, M., den Hartog, F., Sitnikova, E.: Targeted ransomware: a new cyber threat to edge system of brownfield industrial internet of things. *IEEE Internet Things J.* **6**(4), 7137–7151 (2019). <https://doi.org/10.1109/JIOT.2019.2914390>
9. Deorankar, A.V., Thakare, S.S.: Survey on Anomaly detection of (IoT)-internet of things cyberattacks using machine learning. In: 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), pp. 115–117 (Mar. 2020). <https://doi.org/10.1109/ICCMC48092.2020.ICCMC-00023>
10. Demertzis, K., Iliadis, L.: A Hybrid Network Anomaly and Intrusion Detection Approach Based on Evolving Spiking Neural Network Classification. In: Sideridis, A.B., Kardasiadou, Z., Yialouris, C.P., Zorkadis, V. (eds.) *E-Democracy 2013*. CCIS, vol. 441, pp. 11–23. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-11710-2\\_2](https://doi.org/10.1007/978-3-319-11710-2_2)
11. Demertzis, K., Iliadis, L., Tziritas, N., Kikiras, P.: Anomaly detection via blockchained deep learning smart contracts in industry 4.0. *Neural Comput. Appl.* **32**(23), 17361–17378 (2020). <https://doi.org/10.1007/s00521-020-05189-8>
12. Gaddam, A., Wilkin, T., Angelova, M.: Anomaly detection models for detecting sensor faults and outliers in the IoT – a survey. In: 2019 13th International Conference on Sensing Technology (ICST), Sydney, Australia, pp. 1–6 (Dec. 2019). <https://doi.org/10.1109/ICST46873.2019.9047684>
13. Cook, A.A., Misirli, G., Fan, Z.: Anomaly detection for IoT time-series data: a survey. *IEEE Internet Things J.* **7**(7), 6481–6494 (2020). <https://doi.org/10.1109/JIOT.2019.2958185>
14. Demertzis, K., Iliadis, L., Bougoudis, I.: Gryphon: a semi-supervised anomaly detection system based on one-class evolving spiking neural network. *Neural Comput. Appl.* **32**(9), 4303–4314 (2019). <https://doi.org/10.1007/s00521-019-04363-x>
15. Anezakis, V.-D., Demertzis, K., Iliadis, L., Spartalis, S.: A Hybrid Soft Computing Approach Producing Robust Forest Fire Risk Indices. In: Iliadis, L., Maglogiannis, I. (eds.) *AIAI 2016*. IAICT, vol. 475, pp. 191–203. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-44944-9\\_17](https://doi.org/10.1007/978-3-319-44944-9_17)
16. InfluxDB OSS 2.0 Documentation: <https://docs.influxdata.com/influxdb/v2.0/>. Accessed 19 July 2021
17. Žunić, E., Korjenić, K., Hodžić, K., Đonko, D.: Application of Facebook's prophet algorithm for successful sales forecasting based on real-world data. *Int. J. Comput. Sci. Inf. Technol.* **12**(2), 23–36 (2020). <https://doi.org/10.5121/ijcsit.2020.12203>
18. Demertzis, K., Iliadis, L., Anezakis, V.: MOLESTRA: a multi-task learning approach for real-time big data analytics. In: 2018 Innovations in Intelligent Systems and Applications (INISTA), pp. 1–8 (July 2018). <https://doi.org/10.1109/INISTA.2018.8466306>