Utilizing Deep Learning to Identify Potentially Dangerous Routing Attacks in the IoT

Harshith Desamsetti

Senior Software Engineer, Charter Communications, St Louis, Missouri, **USA**

*Corresponding Contact:
Email: harshithdesamsetti9@gmail.com

Manuscript Received: 14 September 2022
Revised Submission: 05 November 2022
Article Accepted: 15 November 2022

ABSTRACT

Due to the rapid increase of cyber-security difficulties brought about by sophisticated assaults such as data injection attacks, replay attacks, etc., the design of cyber-attack detection and control systems has emerged as an essential subfield within cyber-physical systems (CPSs) during the past few years. The outcome of these attacks could be a system failure, malfunctioning, or other undesirable effects. Consequently, it may be necessary to implement the cyber defense system in preparation for impending CPSs to have an improved security system. The various cyber-attack detection schemes based on deep learning algorithms have been intended to detect and mitigate the cyber-attacks that can be launched against CPSs, smart grids, power systems, and other similar infrastructure. This article comprehensively reviews several different deep learning algorithms suggested for use in CPSs to accomplish cyber defense. In the beginning, several methods devised by earlier academics are analyzed in great detail. After that, a comparison study is performed to determine the shortcomings of each algorithm and offer a recommendation for how further improvements to CPSs might be made more effectively.

Keywords: Deep Learning, Artificial Intelligence, IoT Applications, Neural Networks

INTRODUCTION

CPSs connect physical systems for mission-critical operations. Examples include power grids, water distribution plants, and driverless automobiles. These systems are usually combined for remote monitoring and control. Control system communication networks have improved, so CPSs support protection and defense requirements. Internet-connected systems are vulnerable to cyberattacks. Cyberattacks against Ukraine power plants in 2016, Stuxnet's attack on nuclear power plants, and Australia's Moochy water services insider risk in 2000. Thus, industrial task vulnerabilities can seriously impact the economy, public safety, and individual lifestyles, making CPS security essential. Attack detection and control strategies have garnered significant attention in recent decades (Khadija et al., 2017). Many solutions target specific types of attacks, focusing on detection techniques or building resilient controllers based on their characteristics, such as denial-of-service (DoS), false data injection, and replay attacks. However, these strategies can be complicated when they require prior knowledge of episodes to be injected into the system. Precise procedures for manufacturing counterfeit measurement data are easier to distinguish once attacks are identified (Mandapuram, 2017b). Detecting attack types is only sometimes necessary, as the main focus is identifying and eradicating the threat to ensure protection. Systematic procedures have been developed to detect assaults and determine appropriate security rules (Bodepudi et al., 2019).

Many academics are interested in CPS anomaly detection using deep learning. However, such designs and algorithms have limitations, making identifying and removing numerous cyberattacks challenging. This article reviews deep learning-based CPS anomaly detection research. This research examines CPS anomaly detection deep learning techniques. To identify CPS attacks more efficiently, their limitations are addressed.

IoT System Overview and Concept

Today, many things are connected to the Internet to incorporate the concepts of the IoT in various industries, including intelligent buildings, public transportation, medical facilities, industrial facilities, agricultural facilities, and so on(Wortmann & Flüchter, 2015). Specific structures need to be adjusted while activities are in progress to supply customers with features and characteristics of IoT systems and to consider changes in the surrounding environment. The Internet of Things model transforms the conventional entities inside these domains into intelligent ones. IoT is an abbreviation for the Internet of Things, which refers to all the objects linked and continuously associated with one another, such as an electronic device that consists of sensors, actuators, and a microprocessor-embedded module. Because things have to be in contact with one another, requiring Machine-to-Machine (M2M) communication for short-range wireless technology such as WiFi, Bluetooth, and ZigBee, the contact range is either limited or broad, concerning the air and mobile networks for long-distance, including WiMAX, LoRa, Sigfox, CAT M1, and NB-IoT, GPRS, and GSM, and LTE. The Internet of Things aims to provide physical objects with digital identities so that they can interact with one another, exchange information, and get access to a variety of services. The idea that a large number of devices each have their own unique digital identity is a factor that contributes to the development of contemporary Radio Frequency Identification (RFID) technology. These networks were designed to be low-powered machines because of the constraints placed on their resources; this was the impetus for developing resource-constrained Wireless Sensor Networks (WSNs). A highly networked gadget that can be reprogrammed in response to changing needs is an example of an environment enabled by the Internet of Things (IoT). The Internet of Things (IoT) has been utilized to continue patient rehabilitation by adhering to particular criteria, and it has also been used to monitor the parameters that are relevant to the patient (Bodepudi et al., 2021). In addition, the acquired results can be utilized in research comparing patient exposures to various care settings worldwide. The Internet of Things can be used for multiple purposes, including monitoring and managing energy consumption and providing entertainment in agriculture and food production (Dekkati et al., 2022). It can measure and contain meteorological, sociopolitical, climatic, agricultural, gastronomic, and animal illness variables. There is a growing demand for Internet of Things facilities and goods to meet the requirements of an expanding patient population suffering from life-limiting and long-term illnesses and physical ailments.

INTERNET OF THINGS USES

People stand to gain many valuable benefits from the Internet of Things, including simplifying their daily lives and increasing their safety and performance assurance (Desamsetti & Mandapuram, 2017). It has a wide range of applications, some of which include medical equipment, architecture for cities and towns, domestic applications, the design of vehicles, the distribution of electricity, and the "smart world." As a result of the development of ever-more-rapid and sophisticated technology, there are currently a thousand apps for every facet of existence.

- **The healthcare system:** New treatment strategies have been created to improve patients' health. Wireless tracking of wounds can display information without requiring contact with the patient's skin. Other sensors can capture various details, such as temperature, blood oxygenation, sugar, and heart rate.
- **Home Automation System:** This community includes standard household use, such as refrigerators, washing machines, and LED lights (Ángel et al., 2014). These devices were built with access to the Internet to assist in tracking and controlling equipment and optimizing energy usage with one another or registered users. In addition to traditional electronic devices, current innovations are becoming increasingly widespread. These innovations include clever house helpers, sophisticated door locks, and similar products.
- **Intelligent Transport Systems:** It is possible to provide personal assistance, save money, and reduce carbon emissions with the help of sensors that can be built into cars or attached to devices in the city. These sensors can allow for more intelligent route guidance, reserved parking, and communications regarding traffic conditions, telematics, and accident avoidance.
- **Monitor Environment and Conditions:** Any wireless sensor installed in the city can function in various environments (Amin & Mandapuram, 2021). The availability of several barometers and humidity sensors creates more complex weather stations. Intelligent sensors are beneficial for evaluating air quality and water emission levels in the city because they can track pollution both at a distance and on a molecular level. This makes intelligent sensors helpful in assessing the city's air quality and water emission levels.
- **Supply-chain and logistics management:** By utilizing RFID tags, the product's availability during manufacturing and at the store may be substantially decreased, reducing the overall cost and the necessary time. Additionally, it is vital for intelligent packaging to have characteristics such as product authentication, customer quality assurance, client relations, and customization.
- **Protection and Monitoring Devices:** Video cameras with a large storage capacity can receive input from far away. It is feasible to rapidly identify things by utilizing various sensor technologies, which opens the door for developing intelligent protection systems to shield users from potential threats. The many different uses of the Internet of Things may be seen, which covers several life zones (Liu et al., 2017).

A cascade of numerous layers, each possessing a non-linear transformation, is the basis for using the DL algorithmic approach (Dekkati & Thaduri, 2017). The development of artificial intelligence requires the utilization of a wide variety of algorithms, including regression, classification, clustering, auto-encryption, and many more (Desamsetti, 2016a). The sigma neuron is the most fundamental and fundamental logistical node in computation. Deep learning has shown a very high capacity for mapping vast, intricate information into intelligence that is both informative and actionable, which has been a tremendous help in the development of tailored systems (Desamsetti, 2016b). Grouped into two categories: uncontrolled learning, which refers to models with unlabeled data, and controlled learning, which refers to models trained with labeled data. Both types of knowledge are compliant with classical computer instruction. Boltzmann Restricted Machines, often known as BRMs, and Autoencoders are the two types of learning structures that are not controlled. Neural Networks, both convolutional and recurrent, are used as supervised learning models.

USE OF DEEP LEARNING ALGORITHMS IN IOT

A cascade of numerous layers, each possessing a non-linear transformation, is the basis for using the DL algorithmic approach. The development of artificial intelligence requires the utilization of a wide variety of algorithms, including regression, classification, clustering, auto-encryption, and many more. The sigma neuron is the most fundamental and fundamental logistical node in computation. Deep learning has shown a very high capacity for mapping vast, intricate information into intelligence that is both informative and actionable, which has been a tremendous help in the development of tailored systems. Uncontrolled learning, also known as models with unlabeled data, is one of the two categories that make up DL (Zhang et al., 2018). The other category is controlled learning, which adheres to the standards of conventional computer training. Boltzmann Restricted Machines, often known as BRMs, and Autoencoders are the two types of learning structures that are not controlled. Neural Networks, both convolutional and recurrent, are used as supervised learning models. Deep Learning Common Algorithms are:

- **Convolutional Neural Network (CNN):** It is a deep feedforward collection that is based on an Artificial Neural Network (ANN), and it is used for assessing visual imagery (but not recurrently). These networks comprise neuronal nodes with weights and perceptions that can be acquired through experience (Chen et al., 2019). These are the inputs that are received by each neuron, and following that, the neuron produces a dot product. A CNN accepts the two-dimensional entry as a picture or a voice sign and generates features using a chain of hidden layers. The structure is made up of layers that have been compacted and gathered together for extraction, and the two layers that are related to one another serve as classification. CNN may also be utilized in agricultural contexts, such as the diagnosis of cropping and plant leaf diseases, the type of field coverings, the identification of plants and weeds, and the counting of fruits.

- **Recurrent Neural Networks (RNN):** The successive layers constitute what can be thought of as a neuron-like node network (Ballamudi et al., 2021). The soil cover designation, the crop production estimate, the climatic forecast, the calculation of soil humidity, animal science, and a few other things are all linked with a single side-linking node in multiple agricultural regions in the subsequent sequential sheet of each node. Processing data derived from time series is one of RNN's strong suits.

- **Generative Adversarial Networks (GAN)** are fundamentally divided into two competing paradigms of neural networks. These models can be applied to examine the training dataset, analyze it, and replicate its behavior. GAN was utilized on occasion to enhance database quality. These neural networks, one of which is generational and the other of which is discriminatory, work together to provide high-quality output. Both of these networks cooperate. GAN is yet another classification of the ANN, and its use in image processing has been recognized as beneficial.

- **Long- Short-Term Memory (LSTM):** This algorithmic approach is the most efficient. This can handle individual data sources (like the picture) and complete data sequences. In agricultural applications, it is applied for product classification, the prediction of crop production, and weather forecasting. LSTM is frequently used to allude to recognizing handwriting, acknowledgment of expressions, and various other generalized and unique processes for working with DL algorithms (Gutlapalli, 2016b).

Advantages of Applying Deep Learning in IoT Systems

Using DL in IoT systems is an excellent example of some of the possible benefits that may be gained from employing DL algorithms when creating intelligent systems, particularly for this subsector of the IoT market:

- Deep learning requires more complex frameworks for neural networks and can eliminate and define more sophisticated and unknown elements (such as temporal and spatial limits). Deep learning is an alternative to the more traditional, basic ways of research, and it can generalize the intricate linkage of vast amounts of raw data in various IoT applications.
- Deep learning has the potential to enable the most effective utilization of enormous and precious data tools. In most cases, one's ability to comprehend data is contingent upon the breadth and complexity of the experimental study, such as coevolutionary structures. Various information can easily support simpler learning models, whereas more complex learning models can be accomplished by utilizing enormous databases.
- Data-driven learning is a form of end-to-end learning technique that automatically recognizes significant facets of the data without the need for the laborious and time-consuming building of particular functions.

Routing Attack Detection with Deep Learning

With numerous hidden neural network layers, deep learning outperforms shallow learning, which only has one hidden neural network layer. Deep learning is more effective with larger data sets and adaptive to shifting settings (Gutlapalli, 2016a). IoT systems generate enormous data sets, and assaults on them have variable features that linear algorithms cannot identify. However, problems include overfitting25 and the "no free lunch theorem"26, which states that a solution for one problem is unhelpful for another (Valliammal & Shaju, 2018). However, deep learning is a promising machine learning technology for IoT attack detection and prevention.

Using the Cooja simulator, our method generates the first wireless traffic packet capture files for benign and attack situations. IoT routing attack characteristics are used to extract features from packet capture files. The dataset index evaluates feature importance to improve learning accuracy (Mandapuram, 2017a). Elements with high and low volumes are deleted to prevent over- and under-fitting during training. Feature normalization speeds up movement by normalizing datasets. Feature preprocessing produces datasets for deep learning algorithms (Deming et al., 2018). The learning technique uses Python modules like KERAS (https://keras.io), Scikit27, and Numpy28. Learning has the IoT attack detection model. We evaluated the model with multiple situations for more precise precision and recall measurements.

IoT Attack Simulation

Network communication data from various IoT situations with and without malicious nodes must be acquired to create a highly generalizable neural network model (Mandapuram & Hosen, 2018). This was done using the Cooja IoT simulator to model IoT network communication scenarios. Cooja and Contiki are cross-layer (application, operating system, and machine code layer) simulation tools16. The Contiki OS and RPL protocol run on simulated network sensors. Contiki lets you load and unload applications and services to simulated sensors29. To simulate each attack, we ran actual sensor code in the Cooja simulator on a private cloud virtual machine with 48 GB RAM and 8 VCPUs. Since simulations from 100 to 1000 nodes require a lot of memory and computational resources, we used the cloud. Contiki 3.0 runs a 64-bit Java Runtime Environment on 64-bit Ubuntu (Lal et al., 2018).

We simulated IoT routing attacks with different network topologies (Mandapuram, 2016). We used Cooja to simulate these scenarios without creating a synthetic dataset because it allows RPL code to run on simulated nodes. PCAP files can be saved from fake network radio transmissions by Cooja. We next convert the PCAP file to CSV using our Python data preparation package. Feature extraction is then performed on the CSV files. Simulated situations resulted in PCAP file datasets. We converted the PCAP file to CSV using Wireshark and used it for preprocessing.

Deep Learning Routing Attacks and Features

RPL30 is a tree-oriented IPv6 routing protocol for 6LoWPAN. Destination Oriented Directed Acyclic Graphs (DODAGs) are created, also known as DODAG trees. Networks have one or more DODAG root nodes as core nodes and a unique DODAG ID for identification. Furthermore, each node has a rank number and routing table based on the other nodes' ranks. The rank number indicates the distance between the node and the root (Valliammal & Shaju, 2018).

The RPL protocol includes three types of control packets: Destination Advertisement Object (DAO), DODAG Information Object (DIO), and DODAG Information Solicitation (DIS). The root node sends DIO packets as broadcasts to establish the DODAG tree. The remaining nodes receive DIO packets and show their routing table by picking their parent node.

They request authorization to connect to the parent node by sending DAO messages. The parent node responds with a DIO ACK message to accept the offer. A new node joins the DODAG tree by sending DIS packets. When a new node joins the tree, all nodes transmit DIO packets to rebuild DODAG (or network topology). Router attacks occur at the network layer (Lal, 2015). The most prominent routing attacks include lowered rank (DR), hello-flood (HF), and version number manipulation (VN).

DR is a traffic misappropriation assault. In this attack, hostile nodes broadcast DIO packets to neighboring nodes to advertise a lower rank than others. Consequently, neighbor nodes adjust their routing path to include the attacker node by issuing DAO packets (Thaduri et al., 2016). DR attacks can prepare for black holes, eavesdropping, and sinkhole attacks. In the illustration, Node 1 is the DODAG root node, while the others are conventional nodes. The attack does not affect nodes 3–8. Malicious node 9 conducts the DR attack. The attack partially impacts nodes 10 and 11. Some packets from these nodes may be received by the malicious node due to its routing table, causing other nodes to transmit their packets through it. Communication is relayed across the malicious node to the victim nodes (12-18).

The preprocessing of the data and the extraction of the features

For the DR, HF, and VN attacks, we have devised a variety of attack scenarios at a large scale number of IoT nodes, ranging from ten up to 1000 nodes, with a variable proportion (5%, 10%, 20%, etc.) of malicious nodes. These scenarios were tested with a wide range of numbers of IoT nodes (Lal, 2016). Raw datasets were obtained as a direct consequence of running the simulations.

After the simulation, we are presented with raw data files. The raw data files, however, are not enough for use as input to the learning algorithm (Thodupunori & Gutlapalli, 2018). This is because the raw dataset includes information such as the addresses of the source and destination nodes and the packet's length, which contribute to noise and overfitting in the learning process (Koehler et al., 2020). As a result of these considerations, we developed an algorithm for preprocessing data and extracting features in Python by utilizing the Pandas33 and Numpy28 libraries. These libraries perform the essential preprocessing steps to simplify the feature extraction procedure (Tien, 2017).

To manage a significant number of nodes, we developed and implemented a structure called a dictionary (Thaduri & Lal, 2020). Because this kind of calculation could negatively impact the weight analysis for extracted features, we decided not to generate global statistics over the complete amount of simulated time or the total number of packets. We have broken the entire simulation into time frames, sometimes known as windows, each lasting 1000 milliseconds.

Before continuing with this process, sorting the datasets according to the simulation time is required. This is necessary because accurate feature value computations require a correct sequence of packet simulation time. The formulae presented here are used as a guide in computing the values of the features (Gutlapalli, 2017a). The data pretreatment and feature extraction algorithm's pseudocode may be found in Algorithm 1, which also contains the algorithm.

Normalization of the Features

The efficiency of the learning algorithm is hindered because the mean and variance of the data generated by the various IoT routing attack scenarios are uniquely determined by their respective network topologies. To do this, a process of feature normalization is carried out. Both the quantile transforms and the min-max scaling were applied to the datasets we were working with 27. The distribution of feature values is brought closer to a normal distribution using quantile transformation (Gutlapalli et al., 2019). Its goal is to lessen marginal values' impact on the whole. After that, we use min-max scaling to bring all of the values in the datasets into the range of 0-1 values. The depictions show how the feature normalization method affects the features.

Finally, a dataset for an Internet of Things routing attack type is produced by concatenating the data generated by the various network topologies. Consequently, we have three assault datasets. IRAD is the name of the compilation of all of these datasets.

The data normalization algorithm's pseudocode is presented here for our perusal (Mandapuram et al., 2020).

Choosing among the Features

In machine learning, feature selection is one of the most critical steps (Dekkati et al., 2019). Feature selection is typically applied to the dataset before the machine learning algorithm runs. This is done so that it can weed out useless or only marginally relevant features and select the most useful subset of all available features. It does this by determining the appropriate subset of the data's features and making it usable. The massive amount of data and its awkward format are the two primary obstacles to overcome (Gutlapalli, 2017b). Two dimensions make up a dataset: the number of instances and the number of features. Either one or both of these dimensions could be excessively huge. This enormous volume also brings with it a degree of complication.

On the other hand, datasets are constructed from data that does not contain any characteristics or properties. In this context, accurately modeling the attack's consequences on the network becomes crucial (Reddy et al., 2020). IRAD is generated mainly through processing characteristics of network packets recorded in PCAP format from the Internet or a closed network.

We employed a mix of random decision trees (random forests), histograms, and Pearson coefficient correlation for the feature selection process. We used a variety of randomized decision trees, also known as additional trees, to assess the significance of the attributes extracted. Bagging is the essential concept behind randomized decision trees. Bagging refers to the process of adjusting models that are both noisy and unbiased to get a model with a low variance (Gutlapalli, 2017c). Random decision trees function as a vast collection of decision trees that are independent of one another. Bagging is the essential concept behind randomized decision trees. Bagging refers to the process of adjusting models that are both noisy and unbiased to get a model with a low variance. Random decision trees function as an extensive collection of decision trees that are not correlated with one another. Randomized decision trees allow for the creating of many decision trees, which can then be compared to determine the relative relevance of certain features. To accomplish this, we made use of the RandomForestClassifier function that is contained within the sklearn library. The number of estimators that were used was 100. After that, we pick out the most valuable aspects of the design. When a feature's relevance is high, its effect on other parts is diminished, and it might lead to over-fitting during the learning process. Because the importance of features varies depending on the dataset's content, it is recommended that various feature selection methods be applied to the multiple types of attacks (Assem et al., 2016).

It shows that certain qualities contribute more to prejudice than others. The rates that can be found are preliminary essential measurements. First, the feature with the most significance is eliminated to prevent overfitting, and then the part with the most minor energy is destroyed to avoid inadequate fitting. In this step, the importance ratings are recalculated, the procedure is repeated, the learning algorithm is carried out, and the outcomes are analyzed. The process will continue until the elements that provide the best overall value are identified (Mandapuram et al., 2018).

In addition to this, we collected histograms from the dataset to examine the disparities between 0 and 1 within each feature. A feature is considered relevant for the learning process if the line corresponding to the 0 label in a part is noticeably distinct from the cable connected to the one label in the feature and if it follows the histogram distribution for 0 brands. If this is not the case, the label distributions will be very similar to random noise; as a result, the feature will be meaningless, and the learning algorithm will be unable to make good use of it (Lal & Ballamudi, 2017).

Conclusion

This study demonstrates the potential of deep learning for IoT security. Our paper introduces a Big Data-based technique for detecting routing threats with great scalability and generalization. Our suggested attack detection models successfully see routing attacks (decreased rank, hello-flood, and version number). This research addresses a crucial gap in detecting routing attacks in IoT networks.

Limited datasets and poor data quality are the main challenges in this sector. We generate attack datasets through simulation utilizing actual sensor code and the Contiki-RPL protocol. IRAD databases include up to 64.2 million values, a realistic scale for real-life IoT systems. Additionally, we created IRAD dataset-trained deep neural network models with reasonable accuracy, precision, and recall rates. We achieved up to 99% performance based on F1-Score and AUC test scores.

References

Amin, R., & Mandapuram, M. (2021). CMS - Intelligent Machine Translation with Adaptation and AI. *ABC Journal of Advanced Research*, *10*(2), 199-206. https://doi.org/10.18034/abcjar.v10i2.693

Ángel, A., Marco, A., Blasco, R., Casas, R. (2014). Protocol and Architecture to Bring Things into the Internet of Things. *International Journal of Distributed Sensor Networks*. https://doi.org/10.1155/2014/158252

Assem, H., Xu, L., Buda, T.S. (2016). Machine learning as a service for enabling the Internet of Things and People. *Personal and Ubiquitous Computing*, *20*(6), 899-914. https://doi.org/10.1007/s00779-016-0963-3

Ballamudi, V. K. R., Lal, K., Desamsetti, H., & Dekkati, S. (2021). Getting Started Modern Web Development with Next.js: An Indispensable React Framework. *Digitalization & Sustainability Review*, *1*(1), 1–11. https://upright.pub/index.php/dsr/article/view/83

Bodepudi, A., Reddy, M., Gutlapalli, S. S., & Mandapuram, M. (2019). Voice Recognition Systems in the Cloud Networks: Has It Reached Its Full Potential?. *Asian Journal of Applied Science and Engineering*, *8*(1), 51–60. https://doi.org/10.18034/ajase.v8i1.12

Bodepudi, A., Reddy, M., Gutlapalli, S. S., & Mandapuram, M. (2021). Algorithm Policy for the Authentication of Indirect Fingerprints Used in Cloud Computing. *American Journal of Trade and Policy*, *8*(3), 231–238. https://doi.org/10.18034/ajtp.v8i3.651

Chen, S., Thaduri, U. R., & Ballamudi, V. K. R. (2019). Front-End Development in React: An Overview. *Engineering International*, *7*(2), 117–126. https://doi.org/10.18034/ei.v7i2.662

Daniel, O. (2016). The Internet of Things. *Journal of Democracy*, *27*(3), 176-178. https://doi.org/10.1353/jod.2016.0042

Dekkati, S., & Thaduri, U. R. (2017). Innovative Method for the Prediction of Software Defects Based on Class Imbalance Datasets. *Technology & Management Review*, *2*, 1–5. https://upright.pub/index.php/tmr/article/view/78

Dekkati, S., Gutlapalli, S. S., Thaduri, U. R., & Ballamudi, V. K. R. (2022). AI and Machine Learning for Remote Suspicious Action Detection and Recognition. *ABC Journal of Advanced Research*, *11*(2), 97-102. https://doi.org/10.18034/abcjar.v11i2.694

Dekkati, S., Lal, K., & Desamsetti, H. (2019). React Native for Android: Cross-Platform Mobile Application Development. *Global Disclosure of Economics and Business*, *8*(2), 153-164. https://doi.org/10.18034/gdeb.v8i2.696

Deming, C., Dekkati, S., & Desamsetti, H. (2018). Exploratory Data Analysis and Visualization for Business Analytics. *Asian Journal of Applied Science and Engineering*, *7*(1), 93–100. https://doi.org/10.18034/ajase.v7i1.53

Desamsetti, H. (2016a). A Fused Homomorphic Encryption Technique to Increase Secure Data Storage in Cloud Based Systems. *The International Journal of Science & Technoledge*, *4*(10), 151-155.

Desamsetti, H. (2016b). Issues with the Cloud Computing Technology. *International Research Journal of Engineering and Technology (IRJET)*, *3*(5), 321-323.

Desamsetti, H., & Mandapuram, M. (2017). A Review of Meta-Model Designed for the Model-Based Testing Technique. *Engineering International*, *5*(2), 107–110. https://doi.org/10.18034/ei.v5i2.661

Gutlapalli, S. S. (2016a). An Examination of Nanotechnology's Role as an Integral Part of Electronics. *ABC Research Alert*, *4*(3), 21–27. https://doi.org/10.18034/ra.v4i3.651

Gutlapalli, S. S. (2016b). Commercial Applications of Blockchain and Distributed Ledger Technology. *Engineering International*, *4*(2), 89–94. https://doi.org/10.18034/ei.v4i2.653

Gutlapalli, S. S. (2017a). Analysis of Multimodal Data Using Deep Learning and Machine Learning. *Asian Journal of Humanity, Art and Literature*, *4*(2), 171–176. https://doi.org/10.18034/ajhal.v4i2.658

Gutlapalli, S. S. (2017b). The Role of Deep Learning in the Fourth Industrial Revolution: A Digital Transformation Approach. *Asian Accounting and Auditing Advancement*, *8*(1), 52–56. Retrieved from https://4ajournal.com/article/view/77

Gutlapalli, S. S. (2017c). An Early Cautionary Scan of the Security Risks of the Internet of Things. *Asian Journal of Applied Science and Engineering*, *6*, 163–168. Retrieved from https://ajase.net/article/view/14

Gutlapalli, S. S., Mandapuram, M., Reddy, M., & Bodepudi, A. (2019). Evaluation of Hospital Information Systems (HIS) in terms of their Suitability for Tasks. *Malaysian Journal of Medical and Biological Research*, *6*(2), 143–150. https://mjmbr.my/index.php/mjmbr/article/view/661

Khadija, F., Hassan, S., Ayesha, T., Aisha, D., Zohaib, A. (2017). A Systematic Literature Review on the Security Challenges of Internet of Things and their Classification. *International Journal of Technology and Research*, *5*(2), 40-48.

Koehler, S., Desamsetti, H., Ballamudi, V. K. R., & Dekkati, S. (2020). Real World Applications of Cloud Computing: Architecture, Reasons for Using, and Challenges. *Asia Pacific Journal of Energy and Environment*, *7*(2), 93-102. https://doi.org/10.18034/apjee.v7i2.698

Lal, K. (2015). How Does Cloud Infrastructure Work?. *Asia Pacific Journal of Energy and Environment*, *2*(2), 61-64. https://doi.org/10.18034/apjee.v2i2.697

Lal, K. (2016). Impact of Multi-Cloud Infrastructure on Business Organizations to Use Cloud Platforms to Fulfill Their Cloud Needs. *American Journal of Trade and Policy*, *3*(3), 121–126. https://doi.org/10.18034/ajtp.v3i3.663

Lal, K., & Ballamudi, V. K. R. (2017). Unlock Data's Full Potential with Segment: A Cloud Data Integration Approach. *Technology &Amp; Management Review*, *2*, 6–12. https://upright.pub/index.php/tmr/article/view/80

Lal, K., Ballamudi, V. K. R., & Thaduri, U. R. (2018). Exploiting the Potential of Artificial Intelligence in Decision Support Systems. *ABC Journal of Advanced Research*, *7*(2), 131-138. https://doi.org/10.18034/abcjar.v7i2.695

Liu, X., Zhao, M., Li, S., Zhang, F., Wade, T. (2017). A Security Framework for the Internet of Things in the Future Internet Architecture. *Future Internet*, *9*(3), 27. https://doi.org/10.3390/fi9030027

Mandapuram, M. (2016). Applications of Blockchain and Distributed Ledger Technology (DLT) in Commercial Settings. *Asian Accounting and Auditing Advancement*, *7*(1), 50–57. Retrieved from https://4ajournal.com/article/view/76

Mandapuram, M. (2017a). Application of Artificial Intelligence in Contemporary Business: An Analysis for Content Management System Optimization. *Asian Business Review*, *7*(3), 117–122. https://doi.org/10.18034/abr.v7i3.650

Mandapuram, M. (2017b). Security Risk Analysis of the Internet of Things: An Early Cautionary Scan. *ABC Research Alert*, *5*(3), 49–55. https://doi.org/10.18034/ra.v5i3.650

Mandapuram, M., & Hosen, M. F. (2018). The Object-Oriented Database Management System versus the Relational Database Management System: A Comparison. *Global Disclosure of Economics and Business*, *7*(2), 89–96. https://doi.org/10.18034/gdeb.v7i2.657

Mandapuram, M., Gutlapalli, S. S., Bodepudi, A., & Reddy, M. (2018). Investigating the Prospects of Generative Artificial Intelligence. *Asian Journal of Humanity, Art and Literature*, *5*(2), 167–174. https://doi.org/10.18034/ajhal.v5i2.659

Mandapuram, M., Gutlapalli, S. S., Reddy, M., Bodepudi, A. (2020). Application of Artificial Intelligence (AI) Technologies to Accelerate Market Segmentation. *Global Disclosure of Economics and Business* *9*(2), 141–150. https://doi.org/10.18034/gdeb.v9i2.662

Reddy, M., Bodepudi, A., Mandapuram, M., & Gutlapalli, S. S. (2020). Face Detection and Recognition Techniques through the Cloud Network: An Exploratory Study. *ABC Journal of Advanced Research*, *9*(2), 103–114. https://doi.org/10.18034/abcjar.v9i2.660

Thaduri, U. R., & Lal, K. (2020). Making a Dynamic Website: A Simple JavaScript Guide. *Technology & Management Review*, *5*, 15–27. https://upright.pub/index.php/tmr/article/view/81

Thaduri, U. R., Ballamudi, V. K. R., Dekkati, S., & Mandapuram, M. (2016). Making the Cloud Adoption Decisions: Gaining Advantages from Taking an Integrated Approach. *International Journal of Reciprocal Symmetry and Theoretical Physics*, *3*, 11–16. https://upright.pub/index.php/ijrstp/article/view/77

Thodupunori, S. R., & Gutlapalli, S. S. (2018). Overview of LeOra Software: A Statistical Tool for Decision Makers. *Technology & Management Review*, *3*(1), 7–11.

Tien, J. M. (2017). Internet of Things, Real-Time Decision Making, and Artificial Intelligence. *Annals of Data Science*, *4*(2), 149-178. https://doi.org/10.1007/s40745-017-0112-5

Valliammal, N., Shaju, B. (2018). Deep learning algorithm based cyber-attack detection in cyber-physical systems-a survey. *International Journal of Advanced Technology and Engineering Exploration*, *5*(49), 489-494. https://doi.org/10.19101/IJATEE.2018.547030

Valliammal, N., & Shaju, B. (2018). Deep learning algorithm based cyber-attack detection in cyber-physical systems-a survey. *International Journal of Advanced Technology and Engineering Exploration*, *5*(49), 489-494. https://doi.org/10.19101/IJATEE.2018.547030

Wortmann, F., & Flüchter, K. (2015). Internet of Things: Technology and Value Added. *Business & Information Systems Engineering*, *57*(3), 221-224. https://doi.org/10.1007/s12599-015-0383-3

Zhang, J., Zhang K., Qin, Z., Yin, H., Wu, Q. (2018). Sensitive system calls-based packed malware variants detection using principal component initialized MultiLayers neural networks. *Cybersecurity*, *1*(10). https://doi.org/10.1186/s42400-018-0010-y

--0--