# Detection of Intrusions in Big Data Environments Using Advanced Machine Learning

**ANGALAKUDURU SRINIVASA RAO1, Dr. DODLA SRUJAN CHANDRA REDDDY2**

#1Assistant Professor, Department of CSE, PBR Visvodaya Institute of Technology and Science, Kavali

#2 Professor, Professor, Department of CSE-IOT, PBR Visvodaya Institute of Technology and Science, Kavali

**ABSTRACT_** Recently, the huge amounts of data and its incremental increase have changed the importance of information security and data analysis systems for Big Data. Intrusion detection system (IDS) is a system that monitors and analyzes data to detect any intrusion in the system or network. High volume, variety and high speed of data generated in the network have made thedata analysis process to detect attacks by traditional techniques very difficult.

Big Data techniques are used in IDS to deal with Big Data for accurate and efficient data analysis process. This was proposed by Random forest model for intrusion detection. In this model, we have used ChiSqSelector for feature selection, and built an intrusion detection model by using Random forest classifier on Apache Spark Big Data platform.

We used KDD99 to train and test the model. In the experiment, we introduced a comparison between Random Forest, Decision Tree, Linear Discriminant Analysis, Logistic Regression. The results of the experiment showed that Random forest model has high performance, reduces the training time and is efficient for Big Data.

## 1.INTRODUCTION

An Intrusion Detection System (IDS) is a software tool that utilizes various machine learning techniques to identify potential security breaches in a network or system. It safeguards the network from unauthorized access, which can include internal users. The goal of an intrusion detection system is to construct a predictive model (a classifier) that can distinguish between malicious activity (intrusions/attacks) and legitimate connections

## 2.LITERATURE SURVEY

### 2.1 Title: Intrusion Detection Classification Model on an Improved k-Dependence BayesianNetwork.

**Author: L. Jian, S. P. Rui, Y. Min, H. E. Liang, Z. Yuan, and Z. X. Yang**

The very dynamic and varied environment at the edge of the network makes the network security scenario face significant problems. Edge computing extends typical cloud services to

the edge of the network. Therefore, given the emerging edge computing paradigm, it is of utmost theoretical and practical requirement to carry out research and create a high-precision Intrusion Detection Classification Model (IDCM) for network security.

The enhanced k-dependency Bayesian network (KDBN) structural model is studied in this research. It can properly express the dependence connections among system variables and, by minimising the directed edges of weak dependence, may simplify the Bayesian network structure. This study produces an IDCM based on improved KDBN by including the largest a posterior criterion and a virtual augmentation approach for samples of small sizes.

The experiments demonstrate that the IDCM based on improved KDBN has high efficiency, high detection accuracy, and high stability, which optimally addresses the issues discussed in numerous references, such as low detection accuracy and poor stability, for small categories (U2R and R2L) in the KDDCup99 (10%) intrusion detection data set.

**2.2 Title**: **k-Zero Day Safety: A Network Security Metric for Measuring the Risk of UnknownVulnerabilities**.

**Author**: **L. Wang, M. Zhang, and A. Singhal**

A network security metric may offer measurable data to help security practitioners in safeguarding computer networks by permitting a direct comparison of various security solutions with respect to their relative effectiveness. However, challenges in managing zero day attacks that exploit undiscovered flaws have stymied research on security metrics. In fact, because software faults are less foreseeable than other security risks, they have traditionally been viewed as being immeasurable.

Security metrics are severely hampered as a result, as a more secure configuration would be of little utility if it were as vulnerable to zero-day attacks. In order to overcome this problem, we propose a novel security metric in this study called k-zero day safety. Instead of attempting to rank unknown vulnerabilities, our metric counts the number of such flaws necessary to compromise network assets; a higher count denotes greater security because it is less likely that more unknown vulnerabilities will be available, applicable, and exploitable at the same time.

We explicitly describe the metric, assess its computational cost, develop heuristic algorithms for difficult scenarios, and finally show through case studies that incorporating the measure into current network security procedures may result in knowledge that can be put into practise.

**2.3 Title: A Survey on Network Security-Related Data Collection Technologies**.**Author**:

**Lin, Huaqing ; Yan, Zheng ; Chen, Yu ; Zhang, Lifang**

Network security has been the subject of extensive research due to security risks and economic losses brought on by network assaults, breaches, and vulnerabilities. In most cases, data gathered in a network system can be utilised to reflect upon or to identify security threats. These information are what we refer to as network security data. The ability to detect network attacks and intrusions through the study and analysis of security-related data allows for the further assessment of the security level of the entire network system.

Obviously, gathering security-related data is the first step in identifying network attacks and breaches. Collecting these security-related data, however, presents a number of difficulties in the context of big data and 5G. In this paper, we first provide a brief introduction to network security-related data, including its definition, features, and applications. The requirements and goals for the collecting of security-related data are then presented, along with a taxonomy of data collection systems.

In addition, we assess current network data gathering nodes, methods, and mechanisms and analyse them in light of the suggested needs and goals for high-quality security-related data collection. We examine unresolved research problems in the final section, and we offer proposals for future study possibilities.

**3.PROPOSED SYSTEM**

proposed model In this section, the researchers describe the proposed model and the tools and techniques used in the proposed method. Random Forest, Decision Tree, Linear Discriminant Analysis, Logistic Regression model. The steps of the proposed model can be summarized as follows:

- Load dataset and export it into Resilient Distributed Datasets (RDD) and Data Framein Apache Spark.
- Data preprocessing.
- Feature selection.
- Train ML models with the training dataset.

Test and evaluate the model with the KDD dataset

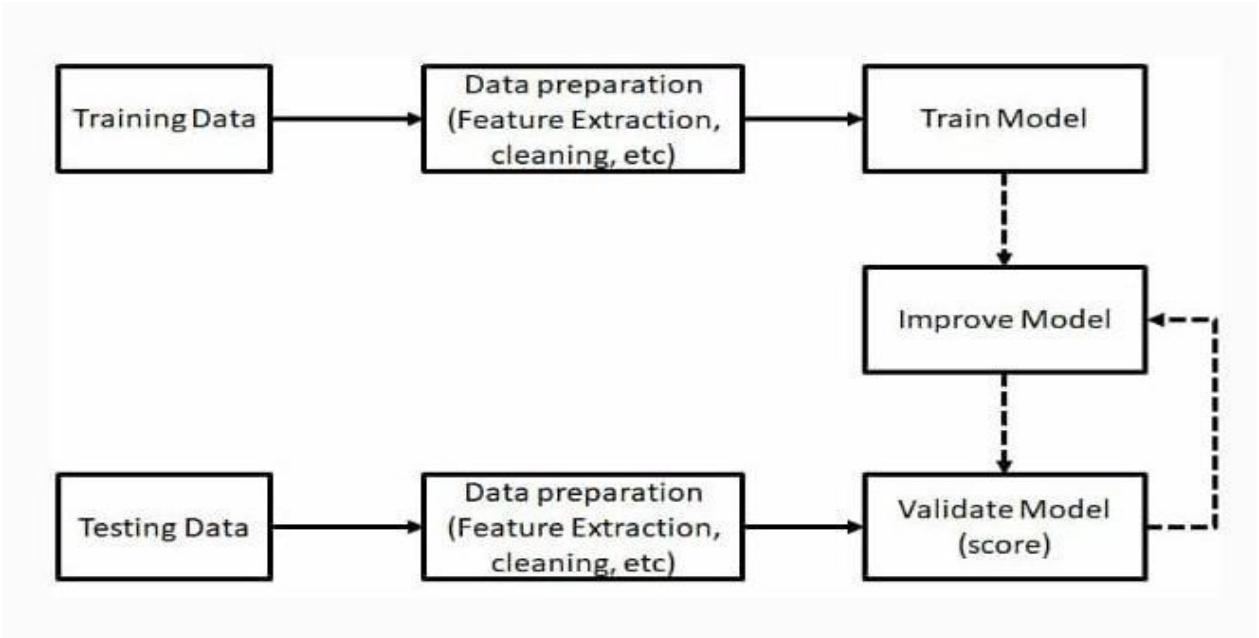


**fig 1:Architecture**

### 3.1 IMPLEMENTATION

- Data collection
- Data Pre-Processing
- Feature extration
- Evaluation model

#### 3.1.1 Data collection

Information is obtained from a variety of sources throughout the data collecting phase, which is then utilised to create machine learning models. The information needs to be kept in a fashion that makes sense for the issue. The data set is transformed in this stage into a digestible format so that machine learning models may use it.

A collection of cervical cancer data with 15 attributes was utilised as the basis for this paper's data. The selection of the subset of all accessible data that you will be working with is the focus of this stage. It's ideal for ML problems to start with a lot of data (examples or observations) for which you already know the ideal outcome. Data that has been labelled indicates that you already know what you want to happen.

#### 3.1.2 Data Pre-Processing

Format, clean, and sample from your chosen data to organise it. Three common data pre-processing steps are:

### 3.1. 3 Formatting

It's possible that the format of the data you've chosen is not one that allows you to deal with it. You could want the data in a relational database or text file if it is in a proprietary file format, or you might prefer a flat file if it is in a relational database.

### 3.1.4 Cleaning

Data cleaning is the process of replacing missing data. There can be data instances that are insufficient and lack the information you think you need to solve the issue. These occurrences might need to be eliminated. Additionally, some of the characteristics can include sensitive data, and it might be necessary to anonymize or delete these attributes from the data altogether.

### 3.1.5 Sampling

There can be far more accessible chosen data than you need to deal with. Algorithms may take significantly longer to perform on bigger amounts of data, and their computational and memory needs may also increase. Before thinking about the entire dataset, you may choose a smaller representative sample of the chosen data that may be much faster for exploring and testing ideas.

### 3.1.6 Feature extration

The next step is to A process of attribute reduction is feature extraction. Feature extraction actually alters the characteristics as opposed to feature selection, which ranks the current attributes according to their predictive relevance. The original attributes are linearly combined to generate the changed attributes, or features. Finally, the Classifier algorithm is used to train our models. On the Python Natural Language Toolkit library, we utilise the categorise module. We make use of the acquired labelled dataset. The models will be assessed using the remaining labelled data we have. Pre-processed data was categorised using a few machine learning methods. Random forest classifiers were selected. The use of these algorithms in text categorization problems is fairly widespread.

### 3.1.7 Evaluation model

The model creation process includes a step called model evaluation. Finding the model that best depicts our data and predicts how well the model will perform in the future is helpful. In data science, it is not acceptable to evaluate model performance using the training data because this can quickly lead to overly optimistic and overfitted models. Hold-Out and Cross-Validation are two techniques used in data science to assess models. Both approaches employ

a test set (unseen by the model) to assess model performance in order to prevent over fitting.

Based on its average, each categorization model's performance is estimated. The outcome will take on the form that was imagined. graph representation of data that has been categorised.

**3.1.8 Accuracy** The percentage of accurate predictions for the test data is what is meant by accuracy. By dividing the number of accurate forecasts by the total number of predictions, it may be simply determined.

## 4.RESULTS AND DISCUSSION

### 4.1 ABOUT DATASET

The KDD'99 Cup dataset is frequently used to assess anomaly detection techniques. The dataset was created using information gathered by Stolfo et al. for the DARPA'98 IDS assessment programme [4]. The information consists of connection records for tcp dump data, each of which is made up of 100 bytes and 41 characteristics. Additionally, each connection is classified as either a regular connection or an assault connection. One of the following four categories best describes the attacks.

- **Denial of Service (DoS):** An attacker makes a memory or computer resource too busy, so that it will be busy to handle a new legitimate request or denies access to a legitimate user.
- **User to Root Attack (U2R):** This form of attack allows an attacker to access a user account on a system whose password is often acquired by dictionary attacks, password sniffing, etc., enabling him to get root access to the system.
- **Remote to Local Attack (R2L**): Attempts are made by the user to get local access from a computer user who does not have access to the system. d. Probing Attack: Attempts to penetrate security measures and gain additional knowledge about a computer network. Think about port scanning. Various Pre-Processing Methods Each connection record hasthe aforementioned traits.
- Fields 0 through 2 include service, protocol_type, and duration. Fields 3 to 40 hold the extra connection details, including the number of failed login attempts, files accessed, and out-of-bounds requests. The 41st field contains the connection type, such as normal or attack. As part of the pre-processing method we used for this data, we removed all non- numeric information from each record and only used the numeric fields for the analysis.

Approach 1: A logistic regression model was utilised. Using training data, we trained the

model. The labels of the test data were then predicted using this model. Table(a) contains a list of the outcomes. Method 2: We utilised feature selection to create a classification model that was more effective. We have employed the correlation between fields for this purpose. In our model, there are several feature selection options..

Different combinations of linked variables can be chosen, or we can choose to ignore some of them. The following combinations of correlated variables have been utilised in our data. • We are aware that the other two fields, src_bytes and servocontrol, which contain information about the number of bytes sent from source to destination and the number of connections to the same service made in the two seconds prior to the current connection, can be used to determine the dst_host_same_src_port_rate. Therefore, we have kept the other two and omitted dst_host_same_src_port_rate..

Similar similarities exist between serror_rate and srv_error_rate. Therefore, we have removed srv_error_rate. The following columns have been removed using the same correlation method. dst_host_same_src_port_rate, column 35. the srv_serror_rate (column 25). the srv_rerror_rate (column 27). dst_host_srv_serror_rate is located in column 38. dst_host_srv_rerror_rate is located in column 40. The aforementioned columns are removed to build a new model.

On the basis of the test results, the accuracy is determined and shown in table (a). Approach 3: Hypothesis testing can be used to assess if a result is statistically significant. An RDD of Labelled Points is used to select features. MLib will do a Chi Square test and internally build a contingency matrix. A few things that don't influence accuracy have been eliminated. After the columns are removed, a new model is produced.

| | duration | protocol_type | service | flag | src_bytes | dst_bytes | land | wrong_fragment | urgent | hot | ... | dst_host_srv_count | dst_host_same_srv_rate | d |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | tcp | http | SF | 181 | 5450 | 0 | 0 | 0 | 0 | ... | 9 | 1.0 | |
| **1** | 0 | tcp | http | SF | 239 | 486 | 0 | 0 | 0 | 0 | ... | 19 | 1.0 | |
| **2** | 0 | tcp | http | SF | 235 | 1337 | 0 | 0 | 0 | 0 | ... | 29 | 1.0 | |
| **3** | 0 | tcp | http | SF | 219 | 1337 | 0 | 0 | 0 | 0 | ... | 39 | 1.0 | |
| **4** | 0 | tcp | http | SF | 217 | 2032 | 0 | 0 | 0 | 0 | ... | 49 | 1.0 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **494015** | 0 | tcp | http | SF | 310 | 1881 | 0 | 0 | 0 | 0 | ... | 255 | 1.0 | |
| **494016** | 0 | tcp | http | SF | 282 | 2286 | 0 | 0 | 0 | 0 | ... | 255 | 1.0 | |
| **494017** | 0 | tcp | http | SF | 203 | 1200 | 0 | 0 | 0 | 0 | ... | 255 | 1.0 | |
| **494018** | 0 | tcp | http | SF | 291 | 1200 | 0 | 0 | 0 | 0 | ... | 255 | 1.0 | |
| **494019** | 0 | tcp | http | SF | 219 | 1234 | 0 | 0 | 0 | 0 | ... | 255 | 1.0 | |

494020 rows × 42 columns

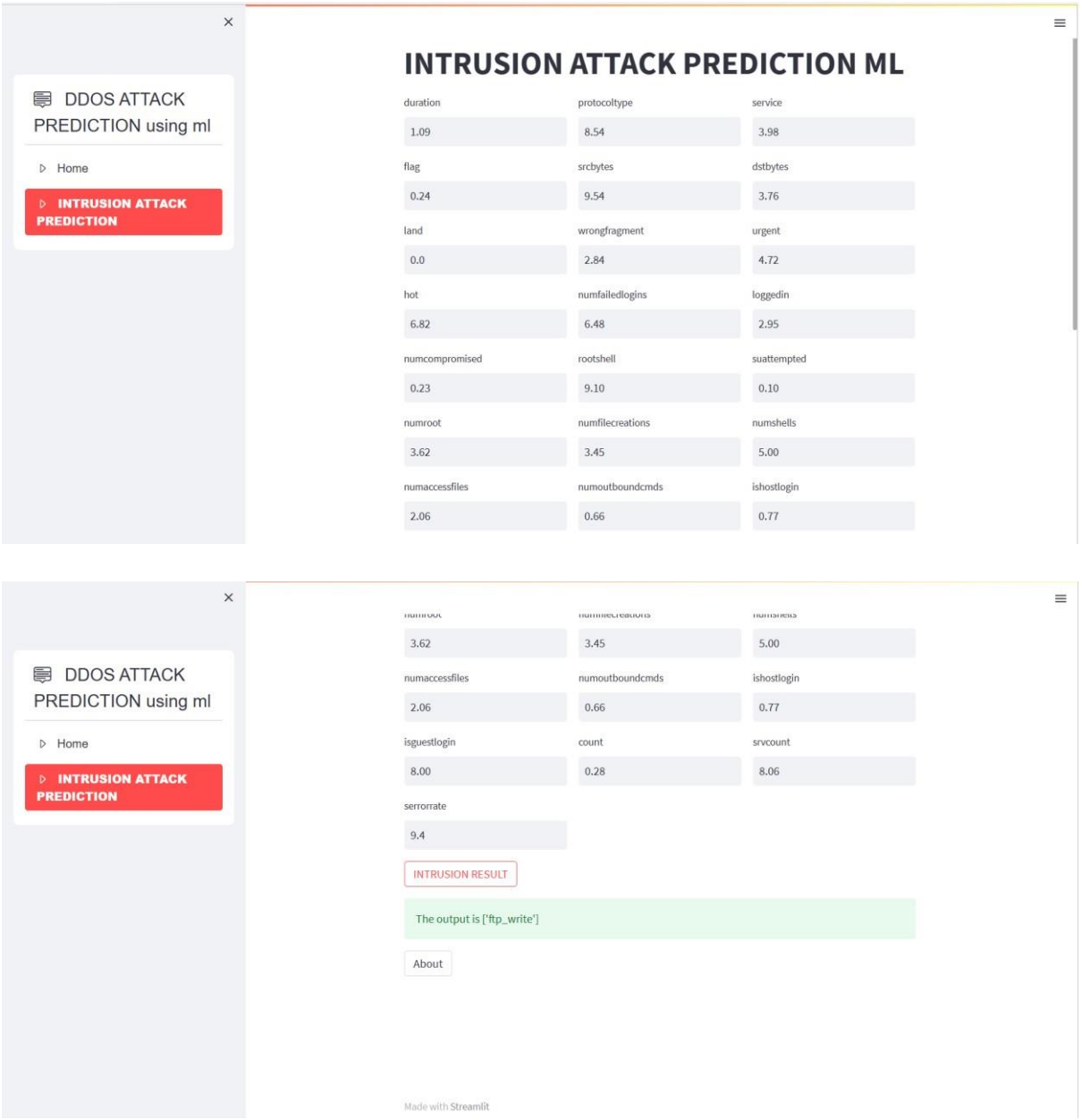


**Fig 2: Expected Output**

The "Strategies" segment shows the recommended model stages and the Flash Large Information equipment that are employed in the executed proposed model to make the model capable of handling Enormous Information. The outcome of the information normalisation approach, which normalises items by scaling them to a unit change, is shown in Table 3. For a few variables selected using the misfeatures strategy, a Chi-selector procedure for selecting highlights, Table 4 displayed the model's findings.

The findings of the analysis model are utilised to compare the suggested model with elective systems.

We performed SVM classifiers without the Chi-selector technique for inclusion determination and Calculated Relapse classifiers with the Chi-selector method using AUROC and AUPR estimates. The results of the analysis demonstrated that the model works effectively and slows the progression of false positives. These are the results with preparation and timing in mind. The suggested model's effects were shown in Figure. After an analysis of the Flash Chi-SVM model and other analysts' planning and forecasting methodologies, the Chi-SVM emerged as the dominant classifier

## 5.CONCLUSION

The analysts introduced Random forest Classifier in this review. The three assault location methods utilized by the IDS are cross breed based recognition, oddity based discovery, and mark based identification. By using the marks of those attacks, signature-based location is planned to recognize known assaults. It is a helpful strategy for recognizing preloaded known assaults in the IDS data set.

Therefore, being much more exact in distinguishing an endeavor at interruption or a realized attack is habitually seen. Albeit the data sets are constantly refreshed to work on their viability of recognition, new types of assaults can't be distinguished in light of the fact that their mark isn't shown.

Abnormality based location, which really looks at current client movement against laid out profiles, is utilized to recognize variant ways of behaving that could comprise interruptions, to address this problem.Without requiring framework refreshes, peculiarity based discovery is powerful against obscure or zero-day attacks.

Tragically, the bogus positive rates for this approach are every now and again significant [5, 6]. To get past the downsides of utilizing only one interruption discovery approach while augmenting the advantages of utilizing at least two, cross breed based location consolidates at least two strategies.

For interruption recognition, many investigations have proposed AI calculations to bring down bogus positive rates and give precise IDS. The commonplace AI draws near, in any case, consume a large chunk of the day to learn and characterize information while managing Enormous Information. IDS can conquer various hardships, like speed and processing time, by utilizing Huge Information approaches and AI.

## REFERENCES

[1] M. P. K. Shelke, M. S. Sontakke, and A. D. Gawande, “Intrusion Detection

System for Cloud Computing," Int. J. Sci. Technol. Res., vol. 1, no. 4, pp. 67–71, 2012.

[2] S. Suthaharan and T. Panchagnula, "Relevance feature selection with data cleaning for intrusion detection system," 2012 Proc. IEEE Southeastcon, pp. 1–6, 2012.

[3] S. Suthaharan and K. Vinnakota, "An approach for automatic selection of relevance features in intrusion detection systems," in Proc. of the 2011 International Conference on Security and Management (SAM 11), pp. 215-219, July 18-21, 2011, Las Vegas, Nevada, USA.

[4] L. Han, "Using a Dynamic K-means Algorithm to Detect Anomaly Activities," 2011, pp. 1049-1052.

[5] R. Kohavi, et al., "KDD-Cup 2000 organizers report: peeling the onion," ACM SIGKDD Explorations Newsletter, vol. 2, pp. 86-93, 2000.

[6] I. Levin, "KDD-99 Classifier Learning Contest: LLSoft s Results Overview," SIGKDD explorations, vol. 1, pp. 67-75, 2000.

[7] J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory," ACM Transactionson Information and System Security, vol. 3, no. 4, pp. 262–294, 2000.

[8] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.

[9] KDD99 dataset, Accessed December 2015, http://kdd.ics.uci.edu/databases/kddcup99

[10] NSL KDD dataset, Accessed December 2015, https://github.com/defcom17/NSL_KDD

[11] P. Ghosh, C. Debnath, and D. Metia, "An Efficient Hybrid Multilevel Intrusion DetectionSystem in Cloud Environment," IOSR J. Comput. Eng., vol. 16, no. 4, pp. 16–26, 2014.

**[12]** Dhanabal, L., Dr. S.P. Shantharajah, "A Study on NSL_KDD Daaset for Intrusion Detection System Based on Classification Algorithms," International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, issue 6, pp. 446-452, June 2015 C. F. Tsai, et al., "Intrusion detection by machine learning: A review," Expert Systemswith Applications, vol. 36, pp. 11994-12000, 2009

**AUTHOR PROFILES**