International Journal of Information Technology and Web Engineering
Volume 17 • Issue 1

SCNTA:
Monitoring of Network Availability and Activity for Identification of Anomalies Using Machine Learning Approaches

Romil Rawat, Shri Vaishnav Vidyapeeth Vishwavidyalaya, India*
Bhagwati Garg, Union Bank of India, Gwalior, India
Kiran Pachlasiya, NRI Institute of Science and Technology, Bhopal, India
Vinod Mahor, IPS College of Technology and Management, Gwalior, India
https://orcid.org/0000-0002-2187-6920
Shrikant Telang, Shri Vaishnav Vidyapeeth Vishwavidyalaya, India
https://orcid.org/0000-0001-5477-865X
Mukesh Chouhan, Government Polytechnic College, Sheopur, India
Surendra Kumar Shukla, Graphic Era University (Deemed), Deharadun, India
https://orcid.org/0000-0003-4120-5953
Rina Mishra, Shri Vaishnav Vidyapeeth Vishwavidyalayaharda, India

ABSTRACT

Real-time network inspection applications face a threat of vulnerability as high-speed networks continue to expand. For companies and ISPs, real-time traffic classification is an issue. The classifier monitor is made up of three modules: Capturing of Packets (CoP) and pre-processing, Reconciliation of Flow (RoF), and categorization of Machine Learning (ML). Based on parallel processing along with well-defined interfacing of data, the modules are framed, allowing each module to be modified and upgraded separately. The Reconciliation of Flow (RoF) mechanism becomes the output bottleneck in this pipeline. In this implementation, an optimal reconciliation process was used, resulting in an average delivery time of 0.62 seconds. In order to verify the method, the authors equated the results of the AdaBoost Ensemble Learning Algorithm (ABELA), Naive Bayes (NB), Decision Tree (DT), K-Nearest Neighbor (KNN), and Flexible Naive Bayes (FNB) in the classification module. The architectural design of the run time CSNTA categorization (flow-based) scheme is presented in this paper.

KEYWORDS

Cyber Security, Machine Learning, Packet Flow, Suspicious Data, Traffic Monitoring

DOI: 10.4018/IJITWE.297971

*Corresponding Author

1. INTRODUCTION

Suspicious mass traffic is constantly evolving, making network behaviour tracing and structure more complex. Cloud-based gaming (Garcia et al.,2021) and grid networks are consuming an increasing amount of SNTA, and flow data is frequently used in traffic monitoring systems.For example, NetFlow (Demertzis et al.,2021) and IETF IPFIX (Goodall et al.,2018) describe a standard for routers and switches to export flow information and are widely used by Internet service providers (ISPs) and businesses to retrieve sensitive business applications, find unidentified signatures, analyze traffic communication patterns, gather data for accounting, and track anomalies. The identification of traffic devices distributed on their networks is a critical concern for companies and ISPs (Xu and Zhu,2021). Semi-supervised learning has received a lot of interest in pattern recognition and ML models. The field of traffic monitoring and categorization has a significant number of journals. The majority of articles concentrate on either reassembling traffic flows or classifying and identifying traffic, but not both. The design of a run-time CSNTA for monitoring organizational networks is described in this document. It also compares and contrasts various ML techniques (Rajawat et al., 2021) for network vulnerable traffic detection. The bidirectional flow principle underpins the classifier monitor. This implies that traffic flow, whether total or subflows, is the fundamental entity to be classified in a determined signature. A flow between two hosts is described by one or more packets of the same quintuple: protocol sort (ICMP, UDP, and TCP), source and destination (Sockets). Deep Traffic analysis (DTA) is the form of Information refining (tracing) (Torabi et al., 2020) that examines about data being transmitted across a network in great detail and takes appropriate measures (like alerting and blocking, rerouting, recording). DTA is frequently used for benchmark application behavior, and monitor network traffic, diagnose network efficiency, ensure for data authenticity and format, and check for suspicious signatures, eavesdropping (Aceto et al., 2021). with network censorship. Despite the description, network infrastructure only has to utilize the first header (the IP header) for regular functioning; nevertheless, usage of 2nd header (TCP/UDP) is typically considered as shallow packet analysis (SPA) (also termed - stateful packet analysis). Packets can be obtained in a variety of methods for DTA. A typical method is to use port mirroring (also known as Span Port) (Torabi et al., 2020) or to physically introduce network tap that copies and delivers data stream at developed for the determination for investigation. DTA (and filtering) allows for sophisticated network configuration, interaction, security features, internet data mining(DM). Despite the fact that DTA used for network configuration for several years, few net neutrality activists are concerned about use of anticompetitive manner or to restrict the accessibility of the Internet. The network telescope (NT) (packets telescope, untrusted network, Network motion sensor, the black hole) (Dias et al., 2019) is indeed a Internet technology allowing users to monitor Huge scale Internet activities. The main idea is to monitor traffic directed at the network's dark (unused) address space. Because all traffic to these addresses is suspect, watching it can provide insight into potential network threats at packet headers (random monitoring worms, DoS/DDoS backscatter), and several further misconfigurations.

Payload-based methodologies, also known as DPI mechanisms (Ahmed et al., 2021), rely on inspecting for the both packet headers and data part to detect any non-compliance with transportation protocols or the presence of spam, vulnerable code, viruses, or security breaches, and then taking preventative measures (blocking, re-routing, or logging the packet). Payload-based systems, on the other hand, cannot handle encrypted traffic since they must match packet contents to static routing rule (Bamakan et al.,2017). DPI techniques also have a large processing overhead, making them unsuitable for real-time use in mission-critical security activities. Although their importance, typical network traffic categorization algorithms can only identify user programmes that run over fixed well-identified network ports like SSH, FTP, HTTP, SMTP, and so on. Most available online users application, on the other hand, make use of dynamic ports, vpn, and encrypted tunnels (Ahmed et al., 2021). Furthermore, many applications use HTTPS links and security protocols (such as SSH and

SSL) to ensure quality of service provisioning, security, and confidentiality. Traditional port-based techniques find it difficult to distinguish such applications as a result.

Now ML, especially DL, had already provided game-changing traffic interpretation capabilities by allowing users to comprehend network traffic behavior and patterns, as well as discriminate between benign and abnormal traffic (Siddiqui and Boukerche,2021). For example, machine learning-based cybersecurity methodologies have contributed significantly in identifying different types of attacks (de Miranda Rios et al., 2021), such as the multi-class distinction of DDOS-attack, DoS-Hulk, the DoS-GoldenEye, Heartbleed, Bot-PortScan, and Web attacks (de Miranda Rios et al., 2021).

1.1 Motivation for the Study

The contributions of this paper are: First, the deployment of an IMT classifier monitoring, Second, the parallel Monitoring are made using the modules (pre-processing with capturing, Reconciliation_of_ Flow (RoF), and categorization), third, analysis and collection of parametric values, Fourth, analysis of classifier monitoring efficiency, Fourth, At classification module the generated result are compared.

The Paper is focused to present the run time architectural design of CSNTA categorization (flow-based) scheme.

1.2 Paper Organization

The Rest of the paper is organized as follows, Section 2 highlight about Related Work; Section 3 Represents about Suspicious Classifier Monitoring ; Section 4 shows about Proposed Approach; Section 5 gives the Outcome And Discussion; and finally section 6 concludes the paper.

2. RELATED WORK

Analytical categorization depends on the premise that each group has a unique distribution of properties to describe and classify it (de Miranda Rios et al., 2021) and is focused on the collection of analytical data based on traffic flow properties. In recent years, mathematical traffic categorization using ML algorithms has received a lot of attention, with a few methods for traffic categorization available in the literature (Li et al., 2007). The NetAI tool can retrieve features both online and offline, but it cannot execute traffic categorization directly. FullStats can derive a large number of characteristics, but only from a limited range of data tracing offline.

The Deep Packet Inspection (DPI)-based program allows for semi-automated tracing and tagging. Tstat 2.0 and TIE are the only two traffic categorization tools that use ML. Tstat uses a Bayesian method to identify Skype and obfuscate P2P file sharing by using packet size and inter-packet time functions. While having a small range of uses, the tool is capable of extracting a vast number of characteristics. The TIE computing framework, which enables the advancement of categorization methods, is open to the scientific community. Traffic (Suspicious Contents) (Salman et al., 2018) capture and sorting, feature extraction, and online categorization are also included in the system. TIE currently has only a limited range of features. When traffic is encrypted, systems like Bro, which can gather flow mechanisms and stats to perform payload-based categorization at high speeds, are constrained (Rawat et al.,2021a).

The commonality of the IP addresses(Ma et al.,2020) became apparent during the research. This is deep indicator about hosts utilizing IP addresses relating to be part of similar subnetwork, belongs towards networks with multiple infected node(Lin et al.,2015) set up for nefarious reasons.

Although the percentage of packets from China(Comar et al.,2013) has decreased, it still accounts for the majority of traffic. On the other side, nations like Taiwan, Turkey, and Colombia experienced an increase in threats and became accountable for higher percentage of the total, while the Country (Germany & Netherlands)(Bamakan et al.,2017) were no longer among top countries

Table 1. Infected IP addresses by Honeynet

Country- Origin	IP address	Packets Count	
Mexico	11x.3x.116.20	187,787	54,170
Vietnam	11x.3x.116.26	1,122,678	467,064
Brazil	11x.3x.116.27	836,785	261,332
Poland	11x.3x.116.21	118,732	52,474
USA	18x.10x.67.248	345,380	70,225
Taiwan	11x.3x.116.8	596,822	165 531
Korea	11x.3x.116.39	521,157	133,996
India	5x.21x.199.181	439,878	132,078
Turkey	11x.3x.116.37	428,583	110,684
Russia	11x.3x.116.4	417,410	106,423
Ukraine	5x.21x.199.218	358,786	94,079
Argentina	11x.3x.116.28	196,210	58,391
Colombia	21x.6x.30.4	225,581	70,225
Romania	21x.6x.30.86	197,997	66,555
China	11x.3x.116.7	4,027,425	590,533

found on previous statistics. The origins of the threats launched against the Honeynet were traced to 172 distinct nations(Dubey et al.,2015). Aside from the many services supplied by each honeypot, the operating system is the most significant distinction between the most targeted honeypot and the rest. The honeypot server, 172.30.20.37(using Windows XP)(Khan et al., 2007), whereas the remainders of threat (honeypots) were running Linux. Despite the fact that research considers this system obsolete, as the recent WannaCry attacks(Moustafa et al.,2018) show, XP is still widely used on many actual networks around the world.

2.1. Stream Reconciliation

The authors of (Mazhar and Shafiq, 2018) proposed an effective TCP stream reconciliation method for high-speed real-time SNTA delivery. To minimise the search cost of a relationship for each packet delivery, the technique employs the principle of (recently-accessed-first). Furthermore, the system holds existing and non-established TCP links in various frameworks to enhance the search operation. Experiments using SNTA in a typical gateway revealed that the proposed strategy was more effective than the older one (RFC 793) (Siddiqui and Boukerche,2021) and meets the run-time property requirements of SNTA systems in gigabit networking.

A TCP stream reconciliation process that was developed and integrated into a network-based intrusion detection system is presented (Mazhar and Shafiq, 2018). Specific packets are sent from the network and signature identification is performed on the payload. The following is a summary of the method: First, based on the quadruple of source destination (Sockets), the device connects every received packet with its associated TCP link. The machine then examines the packet sequence_id to see the intended packet at a given link. The packet (Sivanathan et al., 2018) is then sent for signature identification if correct.

A TCP stream assessment technique that consists of estimating the TCP reconciliation accuracy by precisely identifying possible errors concealed in the mechanism to enhance forensic research is given in Kim et al.,2021. This method can be used to determine and compute reconciliation errors. A

session counting algorithm is provided in the proposed TCP Reconciliation model, which describes the flow of TCP packets along with the same source destination (sockets), and a flow could have several sessions delimited by specified link establishment and termination phases. The libpcap-based software (Yuan et al.,2010) (Abbasi et al.,2021) interprets the packet traces and tests the reconciliation error in verification techniques, and has been experimentally tested with methods (Tcpflow and Tcptrace) based on traffic captures collected with the Tcpdump method.

2.2. Suspicious Traffic Categorization

A binary categorization boosting paradigm suggested by Mazhar and Shafiq (2018) incorporates the benefits of frameworks (graph-based and ensemble approach). The aim is to use unlabeled data to increase the efficiency of a supervised learning algorithm. Semi-Boost is the name of the proposed algorithm, which is a efficient method allowing for the selection of base classifiers suited for particular task iteration. Semi-Boost(similar to other boosting approaches), improves categorization precision by iteration, but it picks up unlabeled data along the way. To achieve the most confident pseudo-labels, proposed techniques blend resemblance knowledge using classifiers forecasting. The authors implemented benchmark semi-supervised methods using WEKA (Gutterman et al.,2019) applications. As compared to 3- state-of-the-art approach of semi-supervised (LDS, TSVM, and the LavSVM), and 16 separate datasets, the proposed solution shows substantial improvement as compared to the base classifiers Decision Stump, J48, and SVM.

A system for traffic categorization was proposed (Siddiqui and Boukerche,2021) that is based solely on ML techniques (Khalid et al.,2018) and packet header knowledge.The suggested architecture includes a mixture of categorization and clustering algorithms to ensure that the recognition mechanism is stable under a variety of network conditions. To make the traffic characteristics diverse, the preparation and assessment of the categorization scheme were done with traffic stream taken from divergent places. When these clustering and categorization techniques were used to classify traffic (Aceto et al., 2019) from unknown networks, the authors discovered that they produced contradictory output outcomes. They also confirmed that clustering algorithms are more resistant to network parameter shifts, while categorization methods learn more precisely about a given network. When compared to standalone cases, the authors describe and test two separate combinations of grouping and clustering methods that result in increased precision.

The first hybrid, called categorization with clustering detail, specifies about every training Stream (flow) has its own cluster id as new supervised categorization features and is determined by clustering the training details taken previously. Because supervised techniques can ignore or give low weight to the clustering knowledge (Chen et al., 2021) attribute, this method cannot always increase global precision. Model refinement with per cluster based categorization is the second method, which uses unsupervised learning to create clusters first. After that, a divergent categorization model is created for each cluster's collection of flows. The unsupervised approach produces most comparable cluster related model and is used for testing an unseen stream during the assessment process. Since each category includes a small number of flow forms, this method often places a high value on clustering performance, and supervised techniques may create basic models.

This means that over-fitting the categorization model (Fotiadou et al.,2021) has less of an effect and per-cluster outperforms all supervised and unsupervised processes on their own. The suggested approaches received TP ratios of 93 percent and 75 percent, respectively, for testing at same network and other networks (for cross-checks).

A semi-supervised predictive traffic categorization technique (Rawat et al.,2021a) suggested handling both known and unknown implementations. The authors state that their proposed semi-supervised approach has three major advantages: Initially, a training(dataset) having a limited count of marked and unlabeled flows, can be used to build quick and accurate classifiers. Second, the method is versatile, allowing it to accommodate previously unknown applications as well as emerging trends in current applications.

Furthermore, network operators may inject unlabeled flows to increase the efficiency of the classifier, allowing for iterative growth. Finally, the proposed method can be combined with flow stats collection solutions. In two steps, the semi-supervised model combines supervised and unsupervised methods: The method begins by partitioning a test dataset using the K-Means clustering strategy. The second stage uses the available labeled flows to create a cluster with application mapping, while clusters without labeled streams remains unmapped, indicating flows may not be associated with any identified applications. The proposed methods and model could correctly classify a wide range of applications (P2P, FTP, Web, and e-mail) .The Flow and byte precision were (98 and 93) percent, respectively. The datasets having wide count of flows reliably attain good categorization accuracy. Despite the tagging methods, the authors confirm that tagging a huge dataset could be costly and complex. In fact, tagging a small percentage of training stream (flows) is enough to achieve appreciable levels for precision.

In terms of classification accuracy, research has demonstrated about EL (Ensemble Learning) (Aburomman et al.,2016) approaches better than single classifier methods, both theoretically and practically. Because there are so many incursions in network settings(Gruhl et al.,2015) especially intrusions in new computer technologies, the advantages of ensemble learning classifier approaches in terms of anomaly detection are particularly evident. As a result, several detection methods are necessary for identification. Furthermore, if one of the classifier approaches fails to detect the threat, other classifier techniques may be able to detect it. Homogeneous and heterogeneous ensembles(Santos et al.,2013) having differing topologies structures. In homogeneous ensembles methods, each classifier are produced using the same approach, but in heterogeneous ensembles, different classifier methods are used. Bagging and boosting, commonly utilized to produce homogenous ensembles, and stacking and voting applicable for building heterogeneous ensembles.

3. SUSPICIOUS CLASSIFIER MONITORING

The configuration and function of our classifier monitoring are described in this section, which is followed by a presentation of the system's modules.

3.1. Architecture

Using collection and preprocessing Package (module), Reconciliation_of_Flow (RoF), attribute extraction and categorization, the display functions as a three-stage pipeline. The time is split into 30 second cycles for pipeline purposes. This number was selected at random. On each cycle, three parallel processes are running: Capturing_of_Packets (CoP), Reconciliation_of_Flow (RoF) of the previous interval, Capturing_of_Packets (CoP), and Categorization_of_Flow (CoF) for the array, which happens in 2-delay intervals. Further parallel processes are in charge of constantly closing old associations to save memory and computing power during the reconciliation process. This method helps the classifier monitor to respond in 30 + alpha seconds, where alpha is the time required to reassemble the captured details within a scheduled slot.

The monitor performs Reconciliation_of_Flow (RoF), function extraction, and categorization with the quantum (30s of traffic) with the average delay of alpha seconds. At present implementation, the average found value was alpha = **0.62** seconds.

The tracking and categorization system's recording and processing ecosystem is depicted in Figure 1. Essentially, we believe the traffic is replicated through network boundary router towards system-monitored network-interface. An machine processes and categorizes collected data on a regular basis, and then displays the results of the tracking and categorization process.

The layered configuration of the implemented classifier monitor is seen in Figure. 2, with online traffic gathering by network stage, preprocessing of Reconciliation_of_Flow (RoF), extraction/selection of analytical Features (attributes), and flow visualization. Tagging using payload examination or a port based approach (during testing), training using Supervised ML methodology, and categorization as

Table 2. Vulnerability Traffic Analysis Approaches

Methods	Objective(s)	attribute selection	Authors
K-means + KNN(K-nearest neighbors)	IDS(intrusion detection system)	Yes	(Islam et al.,2013)
	Encrypted traffic classification	Yes	(Wang et al.,2018)
	IDS	Yes	(Chadha and Jain, 2015)
SVM(support vector machine) + KNN	Zero-day malware detection	Yes	(Hasan et al.,2017)
SVM + PSO(particle swarm optimization)+ kNN	IDS	Yes	(Pimenta Rodrigues et al., 2017)
DT(decision tree)+SVM	Android malware detection	Yes	(Han et al.,2015)
PCA Approach(principal component analysis) Filtering + Probabilistic SOM (self organizing map)	IDS	Yes	(Bhagoji et al.,2017)
K-Means + NB(Naïve-Bayes) + BNN(Back Propagation Neural-Network)	IDS	No	(Ding and Kolaczyk et al.,2013)
GMM(Gaussian mixture model)+ Density Based Clustering	IDS	No	(Carlin et al.,2017)
HC(hierarchical clustering) + SVM(support vector machine)	IDS	Yes	(Saber et al.,2017)
	IDS	No	(Ramadas et al.,2003)
RF(random forest) + AODE(average one-dependence estimator)	IDS	Yes	(Ma et al.,2020)
DT + NB + ANN(artificial neural network)	IDS	Yes	(Dubey et al.,2015)
NB + KNN	Network Anomaly Detection System	Yes	(Mazhar and Shafiq, 2018).
Triangle Area+ Multivariate Correlations	Denial of Service attack detection	Yes	(Sivanathan et al., 2018)
SVM + DT + KNN	Malware classification	Yes	(Abbasi et al.,2021)
	Unknown malware detection	Yes	(Siddiqui and Boukerche,2021)
SVM + RF + DT	Malware classification	Yes	(Salman et al., 2018)
RF+SVM+NB+KNN	Suspicious apps detection	No	(de Miranda Rios et al.,2021)
ES(expert system)+ FL(fuzzy logic)	Network forensics	Yes	(Santos et al.,2013)
GMMs + PSO + SVM	IDS	Yes	(Gruhl et al.,2015)
FL + GA(genetic algorithm)	IDS	Yes	(Aburomman et al.,2016)
	IDS	Yes	(Bamakan et al.,2017)
	IDS	Yes	(Moustafa et al.,2018)
(Proposed Work) AdaBoost Ensemble Learning Algorithm (ABELA)+ Naive Bayes Approach (NB)+Decision Tree Methods (DT)+ K-Nearest-Neighbor (KNN)+Flexible Naive Bayes (FNB)	Network Analysis for Malicious Traffic Monitoring	Yes	

Figure 1. Environment for Malicious Traffic Acquisition

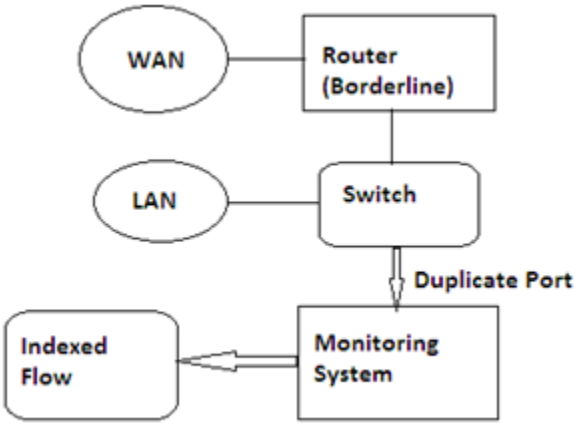
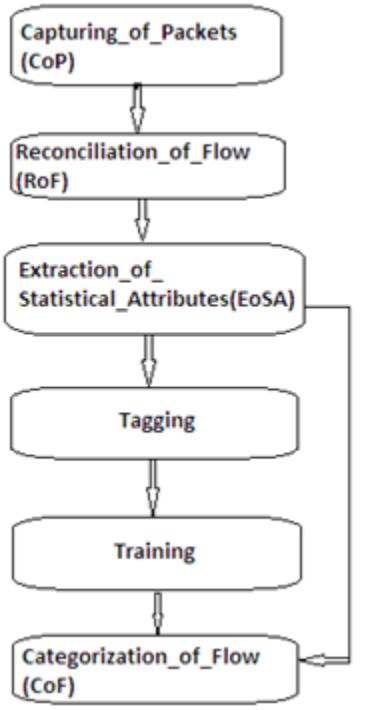


Figure 2. Classifier Monitoring Flowchart



task modules, using a ML model generated from training data. The packet traffic capture is constantly performed by the classifier control. The caught packets are sent to a reconciliation process in the training phase, which associates every packet with its respective flows. An parallel procedure collects analytical Details by Packet-Headers, uses an attribute selection algorithm to pick the most important attributes, and marks the flows using the well-known port system (Demertzis et al.,2021). The traffic flows are used to train a supervised categorization system that is arranged in a spatial presentation

(Every flow is the instance having collection of Attributes). The classifier evaluates the unlabeled flows obtained by compilation, reconciliation, and attribute extraction in the evaluation process.

The data interfacing among every module allow all to be modified and upgraded seperately since modules are applied as concurrent systems. The Packets (TCP) and data structure having required packet Details of monitor are the input and output for Capturing_of_Packets (CoP) and pre-processing, respectively (timestamp, bytes, no-payload, flags,). An input and output for the Reconciliation_of_Flow (RoF) module are a series of preprocessed packets with data-structure that describes the restored flow, respectively. A feature vector (Input) to the categorization module, and class is output.

The display was created using the Visual Studio Integrated Development Environment and the C #.Net programming language. The online Capturing_of_Packets (CoP) is focused on Accessing and processing every packets from an network interfaces in a sequential manner. The control also uses a predetermined timeout for Capturing_of_Packets (CoP) and result presentation. The ability to test various methods for sub-Categorization_of_Flow (CoF), as explained in (Chen et al.,2021), is the justification for implementing a Reconciliation_of_Flow (RoF) Methods, in spite of existence of multiple tools, libraries and packages to achieve the task, like libNIDS (Garcia et al.,2021), TcpTrace (Demertzis et al.,2021), and WireShark (Goodall et al.,2018). Furthermore, as demonstrated by (Xu and Zhu, 2021) and (Pereira et al., 2015), evaluating approaches for run-time TCP stream reconciliation is now possible, which is critical at implementation of high- momentum traffic categorization framework.

3.2. Capturing_of_Packets(CoP) and Pre-Processing

A typical demand for traffic volume control or traffic supervision activities is packet-level traffic capture, followed by data analysis and visualization. As a result, the Capturing_of_Packets (CoP) mechanism must capture completed packets in order for the stream to be properly reassembled. We use the "TCP Session Reconstruct Tool" (Li et al.,2007), a C # utility for capturing_of_packets (CoP) and replication of complete and partial TCP sessions. This method is built on the libnids library (Garcia et al.,2021) and Wireshark and is licenced under the CPOL licence. It employs the TcpRecon TCP restoration algorithm. TcpRecon restores a bidirectional Stream (flow), stores every flow at dictionary format by retrieving the payload for each flow. By replacing the TcpRecon policy with our suggested Reconciliation system, we can reuse this programme.

3.3. Reconciliation_of_Flow(RoF)

A Reconciliation method connects a TCP packet to the stream it belongs to. The aim of this feature is to extract the sender's initial state from the captured TCP packets (Rawat et al.,2021a). It's critical that reconciliation, which can be used in a variety of SNTA analysis systems, including intrusion detection(IDS) and prevention(IPS), content analysis & inspection, and network forensics, done to manage huge traffic volumes, particularly in high-momentum network (Mazhar and Shafiq, 2018). Despite the fact that RFC 973 (Rawat et al.,2021b) provides a common protocol specification (TCP protocol, defined at wide count of RFCs), various execution exist, making TCP Reconciliation challenging. Each reconciliation tool has its own set of stream definition requirements. The Tcpflow method, for example, associates a tuple having source, while Tcptrace(tool) associates the session with stream. The Tcptrace and Tcpflow separate data sent in every direction into separate stream. The data by the sender/receiver is grouped into similar stream provided by the Wireshark tool.

3.3.1. Recently-Accessed-First Principle

Millions of simultaneous interconnections in high-speed networking could be possible, so the Reconciliation method, which keeps a framework for keeping link data, looks for the corresponding record for each collected packet in this structure. If the number of links grows, the search becomes more costly and must be streamlined (Mazhar and Shafiq, 2018). The most-accessed-first theorem aims to put the most recently accessed link documents at the top of the list of interconnection records.

Since data transmission in a TCP link follows the (TCP/IP) specification, an locality theorem applies for packet arrival at the network, based on assumptions that packet with subsequent packet most likely belong to the same interconnection, and that, given the packet arrival, the next packet will arrive soon (Mazhar and Shafiq, 2018). In order to find the communication record collection, the recently-access first principle is used, which is based on the described principle regarding TCP packet arrival. For each active scan, the acquired record transferred at the top of set, ensuring most frequently acquired nodes are at start of the set. As a result, they can be accessed faster, and search performance is enhanced (Mazhar and Shafiq, 2018). While this theory is effective for customers, it has a negative impact on server or interconnection efficiency when there are huge traffic flows, due to loss of the locality theorem in such circumstances (Rawat et al.,2021a). In order to avoid this problem, the concurrent loop in monitor regularly completes old streams (flows).

For optimizing the reconciliation process of our software-based approach, we use the same TCP stream definition as in (Rawat et al., 2021b) and the recently-access-first theory as in (Mazhar and Shafiq, 2018). A single list for storing interconnections is utilized, unlike (Mazhar and Shafiq, 2018) and applies 2-hash tables in link management. The implemented reconciliation strategy focused on the TCP session reconciliation process suggested by Siddiqui and Boukerche (2002). The Tcptrace and Tcpflow tools were used to verify the implemented Reconciliation method. The algorithm for reconciliation operates as follows: In every packet (TCP) obtained, machine scans the link record list for the corresponding interconnection. The kit is put into this one if the record is correct. If record found to be invalid and packet has Flag (SYN), new link is established for packet and it is dropped if the record is found to be invalid and packet doesn't hold Flag (SYN). The link is terminated if packet contains the Flags (RST or FIN).

3.4. Flow Tagging

Tagging is an essential phase in the preparation and assessment of classifiers. While using a port-based approach (Rawat et al.,2021c) to mark traffic flows may induce errors due to growing ineffectiveness, as flows could be labeled incorrectly, an presence of certain imprecise values at datasets is typical ML issue. This is a condition that a successful ML scheme must be able to handle (Nisioti et al., 2021). While this tagging approach was used in the initial prototyping of **CSNTA**, other more advanced tagging methods can be added later.

3.5. Traffic Categorization

ISPs and their infrastructure manufacturers can solve difficult network maintenance issues using run-time internet malicious traffic (IMT) sorting. Network operators, particularly in high-speed networks, must be aware of existing traffic in order to adapt rapidly and support a variety of business objectives (Aceto et al.,2021). The proposed method evaluates NB, FNB, DT, and KNN methods for IMT categorization using analytical knowledge extracted from packet headers using actual traces. Weka (Gutterman et al.,2019) is a JAVA-based Tool(Open Source) that provides an collection of ML methods for Data Mining issues. The libraries (Weka) are used at **CSNTA** to train and evaluate ML methods. To allow Java and.NET interoperability, we use the IKVM (Goodall et al.,2018) programme. This tool helps you to create Weka dlls that can be used in C # code. As a result, weka classifiers can be included in the categorization module of our classifier monitoring.

Data traces (anomaly-free network by edge router) collected at Kasetsart University's (Thailand) (Santos et al.,2013) Internet service centre in the site belongs to students, instructors, and Scholars who want to use the Internet to find useful material for their study. On a daily basis, there are around 1500 users. Users are unable to alter and install tools on the system, and administrators offer necessary tools for all regular users. Furthermore, the administrators keep updated about malicious activities(Wang et al.,2018) and behaviors of the customers up to date on a regular basis. Every day, all operating systems

and applications are automatically reverted to their initial condition, ensuring that they are all clean and anomaly-free. In the training phase, we chose 45 days (clean data traces) (Dubey et al.,2015) for training classifiers, and another 19 days to mix various sorts of vulnerabilities. Chosen anomalies belong to Massachusetts Institute of Technology's Lincoln Laboratory(Moustafa et al.,2018) .These anomalies were made available to researchers who wanted to test for comparing the effectiveness of their own vulnerability detection system. The count of source and destination (addresses, ports, and average packet per second) were the key factors we considered. The ipsweep(surveillance sweep) (Dubey et al.,2015) threat uses port scan or a ping on a large number of IP-addresses. A neptune susceptibility is the SYN-Flood DOS-Threat (Hasan et al.,2017) on traffic streams. To train the classifiers, we require genuine and clean network traffic for developing effective judgment on all classifiers is dependent at training data. Another cause is that the chosen vulnerability (Abbasi et al.,2021) is test data that may be used by anybody to evaluate detection algorithms at traffic(own network).

4. PROPOSED APPROACH

The configuration and function of our classifier monitoring are described in this section, which is followed by a presentation of the system's modules.

4.1. Data Collection and Experiments

Under variable load conditions, the efficiency of the Capturing_of_Packets (CoP) and Reconciliation modules are evaluated for power verification. The classifier display was run on a device (Core i5 having 2.30 GHz Processor, 4GB RAM). A simulation (trace driven) allows for more precision in comparing divergent classifiers and reconciliation methods. Since separate executions of our method for the identical packet trace still produce the same flow sequence, this is the case. Having the risk of delay with packet loss, it will be incredibly difficult to replicate the same effects of an online Capturing_of_Packets (CoP) without this determinism. Traffic traces obtained from host-connected to a broadband Ethernet interconnection (100Mbps) to comfortably test the online monitoring are utilized. A 60-second timeout is programmed for each flow in the reconciliation phase to prevent the storing of idle links, which waste memory computing power. It ensures that flows (TCP) with a lifetime considerable than the value are regularly completed with collector method. Our Reconciliation_of_Flow (RoF) module's time complexity and count of restored flows are compared to the external software Tcptrace, TcpFlow, TcpRecon, and Wireshark.

Www-http- World Wide Web- HTTP, HTTPs (TLS/SSL), FTP, Protocol (Isakmp /Xvttp) were the found application flows in the current traces, as shown in Table 3. (Isakmp Protocol). Www and Ftp implementations are the most common types of K1 traffic traces. The Https and Isakmp applications have more instances in the K2 traffic trace. The categorization and preparation phases are completed at the conclusion of the Capturing_of_Packets (CoP) simulation and reconciliation in our research. The modules of our classifier monitoring must be evaluated using this approach.

4.2. Analytical Attributes

The following characteristics were taken into account when evaluating the categorization process: the time among the first and the last packet, the count of packets and bytes, and each packet having at least an byte of TCP data-payload along with a PUSH bit set at header (TCP), median & variation of number of bytes in an IP packet. Every flow instances having 15 analytical discriminators with class mark since every feature is computed towards both direction of flows (uplink/downlink). We selected some of the most commonly identified features in available work that could be determined from data in packet headers without having to examine their payload.

Table 3. Characteristics of Selected Anomalies

Size_of_Packet		happening, (in sec)	#Packet_Count (Every Second)	%Vulnerability
#Count_Packet	#Bytes (Min: Avg: Max)			
43,729	60:1,292.31:1,514	651	67.16	0.75
43,537	60:1,297.29:1,514	1,064	40.92	1.23
5,658	60:60.26:118	132	42.86	0.15
5,274	60:67.75:118	4,575	1.15	5.3
205,453	60:60:60	3,143	65.37	3.64
460,785	60:60:118	6,376	72.27	7.38
205,607	60:60:60	3,126	65.77	3.62
1,048	60:60:60	1,024	1.02	1.19
1,037	60:60:60	1,015	1.02	1.17
1,606	60:60:60	1,029	1.56	1.19
1,931,274	14:1,066:1,066	1,868	1,033.87	2.16
1,932,328	14:1,066:1,066	1,916	1,008.52	2.22
1,498,077	1,066:1,066:1,066	1,747	857.51	2.02

5. OUTCOME AND DISCUSSION

The performance indicators shown in Table 4 are for the used traces from our classifier monitoring. For the K1 traffic trace, the peak throughput for the Capturing_of_Packets (CoP) and Reconciliation modules was 4.05 flows per second (fps). This indicates the amount of traffic flow delivered per second by the reconciliation method. While this is a poor value, the reconciliation method reaches a throughput of 27458.19 fps at one of the Capturing_of_Packets (CoP) intervals, despite the fact that K1's mean Capturing_of_Packets (CoP) throughput is just 1.29 Mbps. The average Capturing_of_Packets (CoP) and Reconciliation rate was 1179.02 Mbps, measured in Mbits/(Tco+Tre), where Tco and Tre are the combined length periods of CoP and Reconciliation.The same efficiency metrics are seen for the K2 traffic trace. There are no bottlenecks in the reconciliation phase that might impede the considerate traffic flow. For example, our software-based display will function at run time for an organisational network. For K1 and K2 traffic traces, the typical delivery delay alpha is **0.62s** and **8.15s**, respectively. This implies that for a quantum of 32s, this is the average reconciliation time. The distribution delay varies considerably among the 2-compared traces, as can be seen and are distinct, despite the fact that they are from the same Capturing_of_Packets (CoP) point. The K2 has a much

Table 4. Features of the Trace K1

Features	Details
Packet Size mean value	983.81 Bytes
Packets_id	723698
Size of Capture Data	691.32MB
Duration of Capture	4381.81 s
Mean Rate of Capture	1.98 Mbps

Table 5. Features of the Trace K2

Features	Details
Packet Size mean value	1373.27 Bytes
Packets_id	1793728
Size of Capture Data	2.21GB
Duration of Capture	1479.67s
Mean Rate of Capture	12.29 Mbps

Table 6. Categorization of Applications

Categorization and Details	K1	K2
Domain - DNS (Domain Name Server)	N/A	1
FTP - File Transfer Protocol	1938	N/A
HTTPs -TLS/SSL	237	41
Www-http - World Wide Web- HTTP	1098	412
Total	3273	454

higher throughput collection than the K1. Furthermore, K2 has a higher traffic load than K1. As a result, K1 has less traffic to handle and, as a result, has a shorter distribution time than the other traces.

The TcpRecon modified for using 60-second flowing timeout and the count of flows differs between tools due to the previously mentioned divergence of utilized traffic flow approach. Our reconciliation solution takes less time to execute than the other tools. The TCP session Reconstruction Tool was updated to include our Reconciliation scheme, which replaced the TcpRecon default policy. In summary, the implemented recently accessed first theory and use of distinct data structures for holding existing and for not established interconnection (TCP) distinguish these two strategies.

We also equate the performance of TcpRecon and the suggested technique since all written in similar language and the similar Capturing_of_Packets (CoP) libraries. If an occurrence is repeated at least 30 times, the confidence interval estimate of the population would be more reliable [40]. The elapsed times of the above TCP Reconciliation policies were executed and calculated. The policies were assessed using databases that had already been presented. For each TCP regulation, we calculated the average count of execution time and the confidence level. We take a 95% trust level into account. Table 9 shows the resulting values of confidence levels of TcpRecon and the chosen reconciliation

Table 7. Efficiency of Monitoring- Reconciliation Methods

Metric	K1	K2
Throughput (Maximum Capture & Reconciliation)	4.05 fps	2.98 fps
TCP Connections_id	4482	393
Mean Rate of Capture and Reconciliation	1179.02 Mbps	689.17 Mbps
Mean Delay of Delivery	0.62 s	8.15 s
Throughput (Max Reconciliation)	27458.19 fps	172.13 fps
Total Monitoring Time	82.72 s	392.12 s

Table 8. Result evaluation of tools

Approach/Tool	Reconciliation Time	Flows_id
Wireshark	201.13 s	4184
TcpRecon	102.13 s	3813
TcpFlow	131.34 s	6293
Tcptrace	673.56 s	3761
Proposed Approach	86.82 s	4482

Table 9. Efficiency Comparison of Reconciliation Process

Traffic Trace	Proposed Process	TcpRecon
K1	82.14±4.31 s	102.74±5.94 s
K2	397.63±8.31 s	413.83±21.29 s

system. The technique achieves an time complexity benefit of 23.79 sec for K1 traffic-traces. Our solution resulted in an 11.13-second improvement in the K2 traffic trace.

The key findings of the categorization procedure are shown in Table 7. For the two traffic traces, we can see that C4.5-DT correctly categorizes **92.67** percent and **94.81** percent of the traffic, respectively. Utilizing DecisionStump-classifiers, the ABELA became able to correctly classify **83.19** percent and **93.43** percent of the traffic. With k=10, the KNN technique accurately classified **89.77** percent and **97.23** percent of the traffic, compared to **79.39** percent and **86.13** percent for the NB classifier. The categorization process lasted a few seconds, and the findings were used to verify the classifier monitoring's previous phases.

6. CONCLUSION

The design, deployment, and output of an IMT classifier monitoring system are presented in this paper. The monitoring is made up of three modules: pre-processing with capturing, Reconciliation_of_Flow (RoF), and categorization, which were all introduced as parallel procedures. The implementation's throughput reconciliation module for the K1 traffic trace is 27458.19 flows every second. The average time it takes for a package to arrive is **0.62** seconds. The C4.5 (DT) algorithm outperforms the classifiers (ABELA and KNN) in the categorization module, with an average accuracy of **92.67** percent and **94.81** percent respectively, compared to **79.39** percent for KNN and **86.13** percent for AdaBoost Techniques. Incorporating sub-flow-based categorization into **CSNTA** to minimize reaction time is

Table 10. Global Accuracy per Trace

Classifier	K1	K2
Naive Bayes	79.39%	86.13%
K-Nearest Neighbor	89.77%	97.23%
Flexible Naive Bayes	72.84%	92.97%
AdaBoost (DecisionStump)	83.19%	93.43%
C4.5-Decision Tree(DT)	92.67%	94.81%

one of the research's future paths. Second, we want to see how our classifier monitoring affects output on gigabit interconnections, which are becoming more popular in the computer networking domain. And lastly, the **CSNTA** using NetFPGA hardware is prototyped due to difficulty in implementation for any run-time operation, and in gigabit Ethernet.

FUNDING INFORMATION

The publisher has waived the Open Access Processing fee for this article.

REFERENCES

Abbasi, M., Shahraki, A., & Taherkordi, A. (2021). Deep learning for network traffic monitoring and analysis (ntma): A survey. *Computer Communications*, *170*, 19–41. doi:10.1016/j.comcom.2021.01.021

Aceto, G., Ciuonzo, D., Montieri, A., & Pescapé, A. (2021). DISTILLER: Encrypted traffic classification via multimodal multitask deep learning. *Journal of Network and Computer Applications*, *183-184*, 102985. doi:10.1016/j.jnca.2021.102985

Aceto, G., Ciuonzo, D., Montieri, A., & Pescapé, A. (2019). Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges. *IEEE eTransactions on Network and Service Management*, *16*(2), 445–458. doi:10.1109/TNSM.2019.2899085

Aburomman, A. A., & Reaz, M. B. I. (2016). A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Applied Soft Computing*, *38*, 360–372. doi:10.1016/j.asoc.2015.10.011

Ahmed, A. A., & Agunsoye, G. (2021). A Real-Time Network Traffic Classifier for Online Applications Using Machine Learning. *Algorithms*, *14*(8), 250. doi:10.3390/a14080250

Bamakan, S. M. H., Wang, H., & Shi, Y. (2017). Ramp loss K-Support Vector Classification-Regression; a robust and sparse multi-class approach to the intrusion detection problem. *Knowledge-Based Systems*, *126*, 113–126. doi:10.1016/j.knosys.2017.03.012

Bhagoji, A. N., Cullina, D., & Mittal, P. (2017). *Dimensionality reduction as a defense against evasion attacks on machine learning classifiers.* arXiv preprint arXiv:1704.02654.

Chen, C., Liu, B., Wan, S., Qiao, P., & Pei, Q. (2021). An edge traffic flow detection scheme based on deep learning in an intelligent transportation system. *IEEE Transactions on Intelligent Transportation Systems*, *22*(3), 1840–1852. doi:10.1109/TITS.2020.3025687

Comar, P. M., Liu, L., Saha, S., Tan, P. N., & Nucci, A. (2013, April). *Combining supervised and unsupervised learning for zero-day malware detection. In 2013 Proceedings IEEE INFOCOM*. IEEE.

Carlin, D., Cowan, A., O'kane, P., & Sezer, S. (2017). The effects of traditional anti-virus labels on malware detection using dynamic runtime opcodes. *IEEE Access: Practical Innovations, Open Solutions*, *5*, 17742–17752. doi:10.1109/ACCESS.2017.2749538

Chadha, K., & Jain, S. (2015). Hybrid genetic fuzzy rule based inference engine to detect intrusion in networks. In *Intelligent Distributed Computing* (pp. 185–198). Springer. doi:10.1007/978-3-319-11227-5_17

Demertzis, K., Tsiknas, K., Takezis, D., Skianis, C., & Iliadis, L. (2021). Darknet Traffic Big-Data Analysis and Network Management for Real-Time Automating of the Malicious Intent Detection Process by a Weight Agnostic Neural Networks Framework. *Electronics (Basel)*, *10*(7), 781. doi:10.3390/electronics10070781

de Miranda Rios, V., Inácio, P. R., Magoni, D., & Freire, M. M. (2021). Detection of reduction-of-quality DDoS attacks using Fuzzy Logic and machine learning algorithms. *Computer Networks*, *186*, 107792. doi:10.1016/j.comnet.2020.107792

Dias, K. L., Pongelupe, M. A., Caminhas, W. M., & de Errico, L. (2019). An innovative approach for real-time network traffic classification. *Computer Networks*, *158*, 143–157. doi:10.1016/j.comnet.2019.04.004

Ding, Q., & Kolaczyk, E. D. (2013). A compressed PCA subspace method for anomaly detection in high-dimensional data. *IEEE Transactions on Information Theory*, *59*(11), 7419–7433. doi:10.1109/TIT.2013.2278017

Dubey, S., & Dubey, J. (2015, September). KBB: A hybrid method for intrusion detection. In *2015 International Conference on Computer, Communication and Control (IC4)* (pp. 1-6). IEEE. doi:10.1109/IC4.2015.7375704

Fotiadou, K., Velivassaki, T. H., Voulkidis, A., Skias, D., Tsekeridou, S., & Zahariadis, T. (2021). Network Traffic Anomaly Detection via Deep Learning. *Information (Basel)*, *12*(5), 215. doi:10.3390/info12050215

Garcia, N., Alcaniz, T., González-Vidal, A., Bernabe, J. B., Rivera, D., & Skarmeta, A. (2021). Distributed real-time SlowDoS attacks detection over encrypted traffic using Artificial Intelligence. *Journal of Network and Computer Applications*, *173*, 102871. doi:10.1016/j.jnca.2020.102871

Goodall, J. R., Ragan, E. D., Steed, C. A., Reed, J. W., Richardson, G. D., Huffer, K. M., Bridges, R. A., & Laska, J. A. (2018). Situ: Identifying and explaining suspicious behavior in networks. *IEEE Transactions on Visualization and Computer Graphics*, *25*(1), 204–214. doi:10.1109/TVCG.2018.2865029 PMID:30136975

Gutterman, C., Guo, K., Arora, S., Wang, X., Wu, L., Katz-Bassett, E., & Zussman, G. (2019, June). Requet: Real-time QoE detection for encrypted YouTube traffic. In *Proceedings of the 10th ACM Multimedia Systems Conference* (pp. 48-59). doi:10.1145/3304109.3306226

Gruhl, C., Sick, B., Wacker, A., Tomforde, S., & Hähner, J. (2015, September). A building block for awareness in technical systems: Online novelty detection and reaction with an application in intrusion detection. In *2015 IEEE 7th International Conference on Awareness Science and Technology (iCAST)* (pp. 194-200). IEEE.

Hasan, M. S., Dean, T., Imam, F. T., Garcia, F., Leblanc, S. P., & Zulkernine, M. (2017, August). A constraint-based intrusion detection system. In *Proceedings of the Fifth European Conference on the Engineering of Computer-Based Systems* (pp. 1-10). Academic Press.

Han, X., Xu, L., Ren, M., & Gu, W. (2015, November). A Naive Bayesian network intrusion detection algorithm based on Principal Component Analysis. In *2015 7th International Conference on Information Technology in Medicine and Education (ITME)* (pp. 325-328). IEEE. doi:10.1109/ITME.2015.29

Islam, R., Tian, R., Batten, L. M., & Versteeg, S. (2013). Classification of malware based on integrated static and dynamic features. *Journal of Network and Computer Applications*, *36*(2), 646–656. doi:10.1016/j.jnca.2012.10.004

Khalid, W., Ullah, Z., Ahmed, N., Cao, Y., Khalid, M., Arshad, M., ... Cruickshank, H. (2018). A taxonomy on misbehaving nodes in delay tolerant networks. *Computers & Security, 77*, 442-471.

Kim, D., Ho, L., Kim, W. G., Kim, D., & Hwang, D. (2021, February). Poster: A Pilot Study on Real-Time Fingerprinting for Tor Onion Services. *The Network and Distributed System Security Symposium (NDSS) 2021*.

Khan, L., Awad, M., & Thuraisingham, B. (2007). A new intrusion detection system using support vector machines and hierarchical clustering. *The VLDB Journal*, *16*(4), 507–521. doi:10.1007/s00778-006-0002-5

Li, Z., Yuan, R., & Guan, X. (2007, June). Accurate classification of the internet traffic based on the svm method. In *2007 IEEE International Conference on Communications* (pp. 1373-1378). IEEE. doi:10.1109/ICC.2007.231

Lin, W. C., Ke, S. W., & Tsai, C. F. (2015). CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-Based Systems*, *78*, 13–21. doi:10.1016/j.knosys.2015.01.009

Mazhar, M. H., & Shafiq, Z. (2018, April). Real-time video quality of experience monitoring for https and quic. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications* (pp. 1331-1339). IEEE. doi:10.1109/INFOCOM.2018.8486321

Ma, C., Du, X., & Cao, L. (2020). Improved KNN Algorithm for Fine-Grained Classification of Encrypted Network Flow. *Electronics (Basel)*, *9*(2), 324. doi:10.3390/electronics9020324

Moustafa, N., Turnbull, B., & Choo, K. K. R. (2018). An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. *IEEE Internet of Things Journal*, *6*(3), 4815–4830. doi:10.1109/JIOT.2018.2871719

Nisioti, A., Loukas, G., Laszka, A., & Panaousis, E. (2021). Data-driven decision support for optimizing cyber forensic investigations. *IEEE Transactions on Information Forensics and Security*, *16*, 2397–2412. doi:10.1109/TIFS.2021.3054966

Pereira, S. S. L., & Maia, J. E. B. (2015). *ITCM: A Real Time Internet Traffic Classifier Monitor.* arXiv preprint arXiv:1501.01321.

Pimenta Rodrigues, G. A., de Oliveira Albuquerque, R., Gomes de Deus, F. E., De Oliveira Júnior, G. A., García Villalba, L. J., & Kim, T. H. (2017). Cybersecurity and network forensics: Analysis of malicious traffic towards a honeynet with deep packet inspection. *Applied Sciences (Basel, Switzerland)*, *7*(10), 1082. doi:10.3390/app7101082

Rawat, R., Mahor, V., Chirgaiya, S., & Rathore, A. S. (2021a). Applications of Social Network Analysis to Managing the Investigation of Suspicious Activities in Social Media Platforms. In *Advances in Cybersecurity Management* (pp. 315–335). Springer. doi:10.1007/978-3-030-71381-2_15

Rawat, R., Rajawat, A. S., Mahor, V., Shaw, R. N., & Ghosh, A. (2021b). Dark Web—Onion Hidden Service Discovery and Crawling for Profiling Morphing, Unstructured Crime and Vulnerabilities Prediction. In *Innovations in Electrical and Electronic Engineering* (pp. 717–734). Springer. doi:10.1007/978-981-16-0749-3_57

Rawat, R., Rajawat, A. S., Mahor, V., Shaw, R. N., & Ghosh, A. (2021c). Surveillance Robot in Cyber Intelligence for Vulnerability Detection. In *Machine Learning for Robotics Applications* (pp. 107–123). Springer. doi:10.1007/978-981-16-0598-7_9

Rajawat, A. S., Rawat, R., Mahor, V., Shaw, R. N., & Ghosh, A. (2021). Suspicious Big Text Data Analysis for Prediction—On Darkweb User Activity Using Computational Intelligence Model. In *Innovations in Electrical and Electronic Engineering* (pp. 735–751). Springer. doi:10.1007/978-981-16-0749-3_58

Ramadas, M., Ostermann, S., & Tjaden, B. (2003, September). Detecting anomalous network traffic with self-organizing maps. In *International Workshop on Recent Advances in Intrusion Detection* (pp. 36-54). Springer. doi:10.1007/978-3-540-45248-5_3

Salman, O., Elhajj, I. H., Chehab, A., & Kayssi, A. (2018, November). A multi-level internet traffic classifier using deep learning. In *2018 9th International Conference on the Network of the Future (NOF)* (pp. 68-75). IEEE. doi:10.1109/NOF.2018.8598055

Sivanathan, A., Gharakheili, H. H., Loi, F., Radford, A., Wijenayake, C., Vishwanath, A., & Sivaraman, V. (2018). Classifying IoT devices in smart environments using network traffic characteristics. *IEEE Transactions on Mobile Computing*, *18*(8), 1745–1759. doi:10.1109/TMC.2018.2866249

Saber, M., El Farissi, I., Chadli, S., Emharraf, M., & Belkasmi, M. G. (2017). Performance analysis of an intrusion detection systems based of artificial neural network. In *Europe and MENA Cooperation Advances in Information and Communication Technologies* (pp. 511–521). Springer. doi:10.1007/978-3-319-46568-5_52

Santos, I., Brezo, F., Ugarte-Pedrero, X., & Bringas, P. G. (2013). Opcode sequences as representation of executables for data-mining-based unknown malware detection. *Information Sciences*, *231*, 64–82. doi:10.1016/j.ins.2011.08.020

Siddiqui, A. J., & Boukerche, A. (2021). TempoCode-IoT: Temporal codebook-based encoding of flow features for intrusion detection in Internet of Things. *Cluster Computing*, *24*(1), 17–35. doi:10.1007/s10586-020-03153-8

Torabi, S., Bou-Harb, E., Assi, C., Karbab, E. B., Boukhtouta, A., & Debbabi, M. (2020). Inferring and investigating IoT-generated scanning campaigns targeting a large network telescope. *IEEE Transactions on Dependable and Secure Computing*.

Wang, W., Li, Y., Wang, X., Liu, J., & Zhang, X. (2018). Detecting Android malicious apps and categorizing benign apps with ensemble of classifiers. *Future Generation Computer Systems*, *78*, 987–994. doi:10.1016/j.future.2017.01.019

Xu, C., & Zhu, G. (2021). Intelligent manufacturing lie group machine learning: Real-time and efficient inspection system based on fog computing. *Journal of Intelligent Manufacturing*, *32*(1), 237–249. doi:10.1007/s10845-020-01570-5

Yuan, R., Li, Z., Guan, X., & Xu, L. (2010). An SVM-based machine learning method for accurate internet traffic classification. *Information Systems Frontiers*, *12*(2), 149–156. doi:10.1007/s10796-008-9131-2

Romil Rawat is currently working as an Assistant Professor at Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore, India. I have published several research papers and attended many international and national conferences across the world.

Vinod Mahor received the Master of Technology (M.Tech.) degree in Information Technology from Samrat Ashok Technological Institute, Vidisha (M.P), India, Master of business administration (M.B.A.) degree in HR & Finance from ShriRam Institute of Information Technology, Banmore, Morena (M.P.), India and Bachelor of Engineering from Madhav Institute of Technology & Science, Gwalior, MP, India. He is currently working as Assistant Professor in Dept. of Computer Science and Engineering at IPS College of Technology and Management, Gwalior, MP, India, Member of International Association of Engineering (IAOE), Membership of Indian Engineering Teachers Association (IETA) and received short term courses, for attending the research schools related to "online social network security and privacy". The Author has research alignment towards Cyber security, Dark Web Crime analysis, Internet of Things, Investigation techniques and mobile ad-hoc networks including refereed IEEE/ACM/ Springer/Elsevier journals, conference papers, books, and book chapters. His research interests include edge/ cloud computing, the Internet of Things, cyber security, deep learning, artificial intelligence, mobile ad-hoc networks, working towards tracing of illicit anonymous contents of cyber terrorism and criminal activities. Having more than 9 years of teaching and research experience.

Shrikant Telang is currently working as Assistant Professor in Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore, India. He attended several research programs. The Author has research alignment towards Cyber Security, IoT, Dark Web Crime analysis and investigation techniques, and working towards tracing illicit anonymous contents of cyber terrorism and criminal activities. He also joins International Conferences and several research events including National and International Research Schools, Workshops, training programs. He also published several Research Patents.

Mukesh Chouhan is currently working as Lecturer and Head of Department in Computer Science and Engineering at Government Polytechnic College, Sheopur (MP)-India. The author has research alignment towards Web technology, Data Structure and Algorithm, Programming, machine learning, cyber security, Internet of Things. He has published several research papers in referred journals, conference papers and book chapters.

Surendra Kumar Shukla is currently working as an Associate Professor at Graphic Era Deemed to be University, Deharadun, India. He has published several research papers and attended many international and national conferences across the world. He completed his PhD in the area of multi-core architecture. Earlier, he has did B.E. and M.E in Computer Engineering.

Rina Mishra is currently working as an Assistant Professor at Shri Vaishnav Vidyapeeth Vishwavidyalaya, Indore, India. She has research alignment towards cryptography and steganography, and is currently working on cyber crime analysis and detection on an online social network. She attended various research and training programmes and published several research papers.