

Malware Threat Affecting Financial Organization Analysis Using Machine Learning Approach

Romil Rawat, University of Extremadura, Spain*

Sanjaya Kumar Sarangi, Utkal University, Bhubaneswar, India


P. William, Sanjivani College of Engineering, SPPU, Pune-India

Snehil Dahima, SIES College of Management Studies, India

Sonali Gupta, J.C. Bose University of Science and Technology, India

K. Sakthidasan Sankaran, Hindustan Institute of Technology and Science, India

Yagya Nath Rimal, Pokhara University, Nepal

 <https://orcid.org/0000-0003-1045-7728>

ABSTRACT

Since 2014, Emotet has been using man-in-the-browsers (MITB) attacks to target companies in the finance industry and their clients. Its key aim is to steal victims' online money-lending records and vital credentials as they go to their banks' websites. Without analyzing network packet payload computing (PPC), IP address labels, port number traces, or protocol knowledge, the authors have used machine learning (ML) modeling to detect Emotet malware infections and recognize Emotet-related congestion flows in this work. To classify Emotet-associated flows and detect Emotet infections, the output outcome values are compared by four separate popular ML algorithms: RF (random forest), MLP (multi-layer perceptron), SMO (sequential minimal optimization technique), and the LRM (logistic regression model). The suggested classifier is then improved by determining the right hyperparameter and attribute set range. Using network packet (computation) identifiers, the random forest classifier detects Emotet-based flows with 99.9726% precision and a 92.3% true positive rating.

KEYWORDS

Cyber Threat, Emotet, Malware, Online Transaction, Trojan

INTRODUCTION

The number of people using online money lending has increased dramatically in recent years. Because of the growing popularity of online money-lending (Gezer et al., 2019), it has become a target for online deceptive fiscal practices. The amount of malware targeting online device flaws has been gradually increasing in recent years. Cybercriminals employ a variety of tactics to attack online money-lending institutions using fraudulent mails to create malfunctions in users' systems such as phishing emails, key loggers, drive-through downloading, and contaminating targets (victims) with automated and trojanized malware) with the aim of conducting monetary fraud (by botnets, DDOS, data poisoning, and website phishing threats) by capturing user accounts. A monetary botnet is a network of infected

DOI: 10.4018/IJITWE.304051

*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

computers that can be managed centralised by command and control servers (CCS) in order to target monetary customers. Money-lending Trojans are the most devastating threat to fiscal organisations throughout the world and the key drivers of botnet congestion and malignant activities.

When a customer's computer is corrupted with trojanized malware (Ceschin et al., 2019) (Gramatikakis et al., 2021), it transforms into a zombie that can be tracked and even managed by the risk actor. In general, monetary bots identify the following methods to achieve their objectives:

- Insert JavaScript (JS) or HTML into the source code-fragment of targeted websites to track congestion to the updated websites.
- Send the user to a bogus money-lending website that looks just like the real thing.
- Steal data from bank accounts and fiscal organizations.
- To gain additional functionality, the API and plugins

Emotet has been behind MITB attacks since 2014 (Daku et al., 2018), targeting companies in the finance industry by inserting malignant code snippets into existing browser sessions. It first gained popularity in April 2014, when a tailored malvertising campaign targeted corporate and company accounts. Its key goal is to harvest online money-lending details from victims' browsers. Emotet shares a lot of code fragments with the Trojan (Dyre) (Azab et al., 2014), a botnet (Daku et al., 2018) used in a variety of spamming attacks, causing multiples of millions of dollars of damage across the world's leading fiscal institutions.

The accurate detection and prevention of irregularities and congestion found in networks to mitigate or prevent malignant amassment is an essential task in network management. Monetary ransomware is difficult to detect, identify, and test in an automatic manner due to its stealthy nature. A well-identified technique for detecting aberrations in network congestion is ML-based categorization. The detection and prevention of network congestion is usually done using signature or behavior-based methods. A typical series of bytes appearing in a binary-code-snippet is used to classify, detect, and analyse classes of malware in signature-based categorization, but this necessitates scanning of packet payloads. If the packet payload computing (PPC) isn't encrypted, the method described above might be a good way to spot malignant congestion. When analysing (Soomro and Hussain, 2019) user-generated data, this methodology often poses privacy issues and necessitates a lot of computing and storage resources.

For the categorization of emotet-related network flows, researchers have used a behavior-based approach in this analysis along with the objects created by malware during its execution. An ML approach is used to evaluate network congestion flows in an effort to provide some insight into the detection and prevention of an Emotet infection. A test classifier is proposed to discriminate between benevolent congestion streams and malign Emotet congestion streams. Malign congestion captures were created using real-time Emotet malware contaminations, but benign congestion captures were created by interacting with common Internet domains over the defined network. Researchers analysed multiple websites in the Alexa top 600 list to generate benign HTTP congestion during the dynamic analysis of Emotet. The created classifier can be used to identify unknown congestion flows in order to detect any contamination with the Emotet Trojan, and it offers an optimistic method for classification of Emotet-related congestion flows without accounting for network parameter values. Emotet's CCS IPs are constantly modified, and used port numbers are allocated dynamically. As a result, considering parameter values may over-fit the generated classifier, resulting in a biased result in real-world scenarios when researchers encounter new Emotet versions. The Emotet money-lending Trojan is revised on a regular basis, almost every day, and training the classifier with socket variables can result in a skewed throughput that relies on specific sockets. Over the course of ten months, researchers dynamically analysed over 400 Emotet malware illustrations and recorded their network congestion with a Network Monitor. The seized files were then used in Emotet classifiers as both the training dataset and the testing dataset to assess the model's results.

BACKGROUND AND NEED OF CURRENT RESEARCH

- To determine the strength of anomalous patterns for each feature.
- The malware comes as spam email (Malicious macros) to steal the user account and credentials. And most of the time, it goes unnoticed and results in account compromise.
- Frequently, changes in infection techniques and attack vectors occur.

The Novelty and Contribution

- The current research is focused on the Emotet money-lending Trojan class.
- A classifier is created for detecting Emotet Susceptibility based on streams (flows) detected on networks.
- Compared with the output of 4 distinct classifiers.
- Forward Identification (Consistency Size-Subset) The Assessor is used for refining the attribute set.
- Tests were conducted to find the best set of hyper parameters for Emotet flow identification, including 150 categorization trees and four random attributes for tree splitting.
- Over the course of ten months, 400 examples of Emotet malware have been analysed and recorded using Paessler PRTG Network Monitor (intercept network congestion) for Network Scan (Packet Capturing .pcap file). The seized files were then used in Emotet classifiers-the training-dataset and the testing dataset to assess the model's results.
- The Accuracy 99.83% is achieved by training the classifier without any subsampling operations and only relied on TP rate while evaluating our classifier.

The remainder of the paper is organised as follows: Section IV is a review of the literature; Section V is about the Emotet Money-Lending Trojan; Section VI is about the methodology; Section VII is about throughput metrics; and Section VIII is about Comparison with Available Technique and finally section IX concludes the paper with future work.

LITERATURE REVIEW

An aggressive strategy for combating the Zeus botnets, CCS, or bot-herder, was suggested. The Zeus botnet class has been around for over a decade and is one of the most well-identified money-lending Trojans. In 2017, the Zeus botnet and its variants accounted for nearly 29% of all money-lending malware. Monetary malware developers have been motivated to include Zeus functionality in their cybercrime software after the source code-fragment for Zeus was leaked in May 2011. The malware families ICE IX and Citadel are both related to the leaked Zeus source code-fragment. Citadel's primary aim, like Zeus', is to offer online penetration. Complex MITB attacks may be carried out using the revamped websites with injected JS code-snippets or new type fields.

A behavior-based categorization strategy was given by Gezer et al. (2019) to identify the Zeus money-lending Trojan, which relies on objects generated by the malware during execution. They extract 67 attributes that are exclusive and durable for distinguishing malware families for categorization. They prove that design objects' values (file system, network attributes, and registry) can be used to accurately distinguish different malware families, up to 96% in some cases. In this analysis, however, they did not use attribute selection to improve their model. Existing attack tactics and vulnerabilities affecting existing internet money-lending networks are categorised in his work. They offered guidelines for designing stable internet money-lending systems that are not compromised by existing threats and vulnerabilities, based on an in-depth study of current security models. The authors of the above study did not rely on any specific money-lending Trojan in order to expose their specific attacking vector space technique.

Niu et al., 2019 presented the CKC (cyber kill chain) Process designed phylogeny in money-lending Trojan attributes that can be used to warn identification and mitigation strategies. Their phylogeny is focused on the study of 129 money-lending Trojan illustrations collected by incident-based events that happened at monetary institutions in the UK between December-2014 and January-2016. In the report, they did not attach any special signature to any specific money-lending Trojans.

The aim of this paper is to look at how similar the behaviours of this particular set of malware (Casino et al., 2021) are, despite the fact that they all have distinct and somewhat unknown roots. This higher-level abstraction of malware functionality is represented by the automated categorization of the behaviours present in a malware illustration. (Rawat et al., 2021a). presented to Zarathustra. (Kono et al., 2018), a method that automatically characterises web penetration-based actions independently of the underlying implementation. They tested their system on 213 true, live money-lending website URLs and 56 different Zeus Trojan illustrations. None of this research examined the Emotet money-lending Trojan's actions.

ML methods are commonly used to detect specific network congestion flows. A ripper learner is used to detect SSH congestion with a 99 percent detection rate. They remove port (TCP and UDP) numbers from the attributes set to avoid any skewed models that rely on randomly assigned port numbers. (Kono et al., 2018) used ML methods to compare certain flow exporters in order to improve botnet congestion categorization. They tested the Tranalyzer flow exporter with the botnet dataset (Rawat et al., 2021b) (Citadel, Kelihos, Cutwail, Conficker, and Zeus) and found that it generated the best results. They did not, however, conduct any tests with the Emotet botnet.

A bot detection and prevention scheme that uses a multi-layer perceptron classifier to detect randomised bot congestion (Rajawat et al., 2021) for early bot detection and prevention (Rawat et al., 2021b) To distinguish between benevolent and malignant bot congestion, transient diligence that lasts longer when compared to a user session They tested 43 bot illustrations, including Ban-bra, Dedler, and Ramnit, and suggested an ML-driven botnet identification method related to congestion behaviour attributes and flowing segments. They did, however, provide parametric values for the attribute set (IP address, source-destination ports, and protocol details), which may lead to biased models relying on fluctuating socket identification values. A survey was conducted to examine the similarities between money-lending malware habits. They concentrated on compiling a dataset of malware activity using static approaches, as opposed to our research, which used dynamic analysis to identify the Emotet money-lending Trojan.

Based on research, the proposed work is the foundation to focus on detection and prevention of Emotet money-lending Trojan injection and malignant congestion in networks. Researchers used a behavior-based ML approach in this analysis to classify emotet-related congestion flows within a network. Without evaluating the application's PPC or IP addresses, the designed categorization model may detect some Emotet infection.

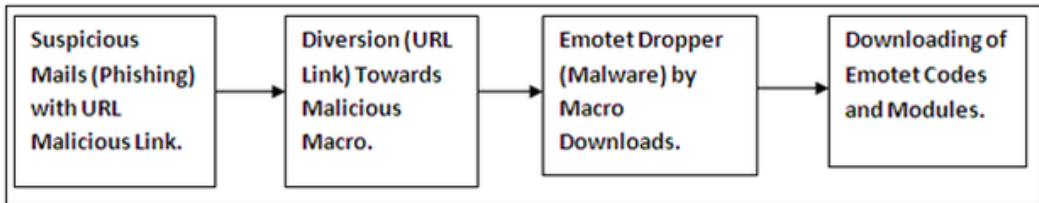
EMOTET MONEY-LENDING TROJAN

Emotet is the world's first and only money-lending Trojan that attacks customers' major banks in a variety of geographies and languages. Initially, Emotet attacked monetary institutions in the United Kingdom and on the Australian continent. Emotet activators are launching redirection attacks against banking securities across 21 divergent countries as of late (Rawat et al., 2021b). The authors of Emotet called it the Emotet loader based on the strings contained within the code-fragment. It has a lot in common with the Dyre money-lending Trojan in terms of internal design-makeup, processes for malfunctioning new hubs, and modular form. It deceives consumers into providing contact details, publicly identifying facts, and monetary authorization code-fragments on fake money-lending websites (Phishing). The Emotet malware has been updated with newer packages (plugins) and customization files on a regular basis, enhancing its ability to steal passwords, extend delivery vectors, and avoid detection and prevention by security protectors. The Emotet contaminates the victim's computer with

a ransomware scrap (binary). The binary then begins to download various packages (plugins) that are responsible for various operations. If browsers are accessed at this stage in the infection, they are then used to insert malign code-fragments into targeted websites. It conducts exceptionally qualified browsing redirection links.

As seen in Fig. 1, the Emotet binary activated by the phishing mail contains malignant links to URL, which if clicked, diverts towards the malign macro and then the Emotet dropper downloads the code-fragment and packages (plugins) of Emotet Malware. It is made up of multiple layers. The first layer serves as a barrier. It transports the encrypted payload and attempts to conceal it from anti-virus applications. The key bot loader is the second layer and decides to execute a payload (32-bit or 64-bit) based on the architecture of the available device. To implement and download the authenticated designed architecture of relevant resource files, this malware first gathers knowledge about the target's OS (operating system) (Daku et al., 2018). The majority of its correspondence with its main CCS is encrypted (Gezer et al., 2019) (Malwarebytes Labs, 2016).

Figure 1. Emotet Activation by Malignant URL links



Mathematical Modeling and Setup

- i. P_e (“myexternalip.com,” “icanhazip.com,” and “ip.anysrc.net”) $\rightarrow IP_s$
- ii. $S \rightarrow W_i$ (dinj and sinj).
- iii. $N_c \rightarrow$ Emotet and Non-Emotet $\rightarrow P_{st}$
- iv. $P_{st} \rightarrow$ DS Classes.
- v. Paessler PRTG $\rightarrow NS \rightarrow PC$
- vi. Scan $E_v \rightarrow$ Folder AppData [“winapp,” “netdefender,” and “services.”]
- vii. $LF_{eip} \rightarrow$ decrypted files(config) .
- viii. $P_{cs} \rightarrow$ Encrypted files retrieval (ServConf and the config.conf).
- ix. Repository file $\rightarrow LF_{eip}$
- x. Classifier Evaluation \rightarrow Attributes (Weka Tool)
- xi. Modeling Construction \rightarrow (10-fold-cross) validation (RF , MLP , Naive Bayes, SMO, and LRM)
- xii. Parameter Evaluation (confusion matrix)

where,

P_e – Process Evaluation

S- Scan

IP_s – IP Address Searching.

W_i – Directory Customization Containing URL.

N_c – Network Class (congestion flows)

P_{st} - Packet statistical Attributes computation

DS- Dataset

NS- Network Scan (Network Monitor intercept network congestion)
PC – Capturing (.Pcap file)
E_v – Emotet Vulnerability
P_{cs} - Code snippet and script (Python)
LF_{eip} - Emotet CCS IP library file

Behavioral Analysis

As the emotet binary activates, it removes code-fragment and stores it in a heap location with the invocation of entry points. Emotet's first operation is to obtain the victim's public IP address, which it then shares with its CCS. Following that, it loads encrypted packages (plugins) and customization files from the botnet's CCS on a regular basis. These files specify how the malware can alter the content of bank webpages that have been attacked. The key file for customization (config.conf) contains details about the version, community tagging, IP trace locations, and labels of CCS after decryption. A list of IP addresses from which Emotet can import relevant resource files can be found in the config.conf file. The contents of the customization file are significant in this sense, since system administrators will restrict access to these IP addresses. The disadvantage of this strategy is that it is reactive rather than constructive. Emotet updates this key customization file on a (nearly) frequent basis, requiring this list of restricted IPs to be revised on a regular basis. Researchers used the list of CCS IPs from Emotet's customization files to mark network data flows for the purpose of testing and training.

WHEN A DEVICE IS CORRUPTED BY THE EMOTET TROJAN, THE FOLLOWING ACTIONS NEED TO BE TAKEN:

- It replicates itself in the AppData percent Roaming-based folder as a subdirectory.
- It generates identification files (client id and group tag) as well as a Packages (plugins) folder for storing different packages (plugins) and inserting customization files.
- Creates a task in the Windows Job Scheduler to maintain stability after a server resets or termination.
- It establishes a connection with a server in order to obtain the target's public IP address.
- Downloads packages (plugins) and customization files from its catalogue of CCS IPs.
- The main process launches many svchost.exe processes (usually between four and seven), each working on various tasks related to the downloaded packages (plugins).
- Code-snippet penetration is done, redirecting towards unauthentic domains.

Both the group tagging and client-id files contain text in Unicode and are not encrypted. the Windows OS's construct edition, and an arbitrarily produced 32-character string. The encrypted plug-and-play packages (plugins) will be stored in the newly developed Packages (plugins) folder. Check for mcconf in the memory-strings of operating malware binary to find the "made in" key customization (similar to the "config.conf" file downloaded later). The string value holds the configuration's version id, community tagging, and a series of CCS IPs. Emotet tries to communicate with these various IPs (apparently at random), but if an active CCS is detected, a TCP link is created to retrieve supplementary base files, including modified main customization (config.conf), and supporting packages (plugins). Table 1 shows Customization File Functionalities. The key emotet method induces malignant (svchost.exe) sub-routines that correspond to all packages (plugins) and their attributes when they are downloaded. Researchers have used the Emotet decoding techniques for decrypting the base files after downloading the Emotet Toolkits.

Table 1. Customization file functionalities

File	Functionalities
systeminfo32 -	Collects details about the infected device.
dpost -	The complex web injects can send captured queries to the server.
sharedll32 -	Allows malwares to migrate laterally across network links by looking for a list of email accounts in personal directories.
injectdll32 -	Embeds site injects into the browser.
mailsearcher32 -	Looks for a list of email accounts in personal archives.
sinj -	Specification of static web injects.
config.conf-	Main Emotet customization file, which contains the most up-to-date server list, edition, and community tag.
dinj -	Functionality of dynamic web insert
mailconf -	Server to which the extracted email list will be sent.

METHODOLOGY

ML Approach for Identifying Emotet Flows

Researchers are primarily interested in detecting Emotet infection and identifying Emotet-related network flows. As the Emotet Trojan contaminates a system, it initiates network operations to public service sites like “myexternalip.com,” “icanhazip.com,” and “ip.anysrc.net” to obtain the target’s public IP-address. Following that, the controlled device attempts to bind to one of the Emotet CCS that is currently running. The malware’s binary contains several hardcoded CCS IPs. It will receive new IPs once the link is created, with the goal of continuously downloading encrypted packages (plugins) and customization files, as well as exfiltrating stolen data. The Ramnit money-lending Trojan, on the other hand, communicates with its CCS using a domain generation algorithm (DGA). The trojan begins performing DNS queries for newly created domain names as soon as it is infected. IcedID, a well-identified money-lending Trojan that uses a very different network connection pattern, creates a local proxy and routes Internet congestion through it.

Emotet makes use of web injects details in its dinj and sinj customization directories, which contain URL links to web injects scraps. Web injects are often JS scripts that are loaded into webpages when users are browsing legal online money-lending sites. The quality of dinj and sinj files is difficult to expose without decryption since they are encrypted with AES. To escape detection and prevention by intrusion detection and prevention systems, the Emotet Trojan downloads packages (plugins) and customization files using TLS/SSL (Rawat et al., 2021a) network communications. Other than 443, it occasionally uses multiple ports for TLS/SSL correspondence. The Emotet takes advantage of virtual-network computation, which allows the invader to take control of the victim’s computer. Emotet also uses screenshots to expose any use of a simulated keyboard to insert user credentials into a money-lending session. The majority of these capabilities are acquired by running downloaded packages (plugins) and customization files from CCS. They often need network connectivity after executing these packages (plugins) in order to achieve their disruptive goals. However, understanding their motivation is problematic due to the encrypted content of shared packets (Computing). As a result, understanding the Emotet network contact pattern will aid us in detecting any Emotet infection on a device. To achieve this goal, researchers concentrate on the coordination patterns between Emotet CCS and the compromised device. In our implementation, researchers use an ML strategy to divide network flows into two categories: emotet congestion flows and non-emotet congestion flows. A series of statistical attributes may be computed from one or more packets (computation)

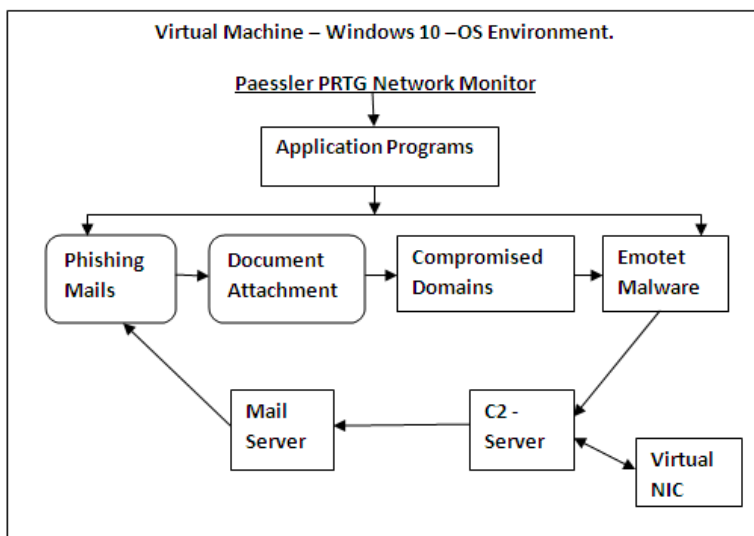
to characterise each congestion flow (Kono et al., 2018). As a result, each flow has the same set of function names but different attribute values. Researchers use a supervised ML approach in our methodology to characterise congestion flows into class memberships. The groups in supervised categorization should be predefined before the machine is trained. To begin, a categorization model is created by a dataset (training) that contains examples of every class. Then, using this model, new congestion flows, described as statistical attributes, are used to forecast class membership.

Data Collection via Dynamic Analysis

Researchers dynamically tested several Emotet malware illustrations from version 2 to version 187 over the course of ten months. Following the execution of the study, network contact takes place between the poisoned computer and a server, revealing the poisoned system address. At the review stage, researchers look for any newly generated folders linked to the Emotet infection in the percent AppData percent Roaming list. According to the Emotet edition, the names of this newly developed folder are “winapp,” “netdefender,” and “services.”

Paessler PRTG Network Monitor intercepts network congestion using pre-designed screening regulations to not siege local broadcast packets (computation) until the emotet malware illustration is executed on a virtual machine. Figure 2 illustrates our implementation of Emotet’s complex analysis. Researchers often use Process Hacker to keep track of the virtual machine’s operating operations. In this way, it’s easy to see when the emotet operation begins, stops, or is employed in a job scheduling environment to execute downloaded packages (plugins) or DLL scripts. Over the course of ten months, researchers studied over **361** separate emotet experiments to see how the Trojan evolved and new behavioural trends emerged. When visiting a money-lending website, is there any web-injection? Is it sending people to some phishing sites? Researchers capture the network congestion using Paessler PRTG Network Monitor to track the relationship between the host and Emotet CCS when performing complex analysis to find answers to these queries. Packet exchange is higher, particularly when downloading customization and packages (plugins) data. Researchers discovered a network exchange sequence and concluded that the pattern can be used to detect Emotet conducts. Researchers purposefully create benign HTTP congestion during the network congestion capture process when the Emotet illustration is running by interacting with common domains. Visit certain well-identified blogs,

Figure 2. Designed framework of Emotet simulation environment



such as academic domains, internet newspapers, and well-identified social media websites (Twitter, Facebook, and Instagram) to attract such innocuous congestion. captured. pcap files contain both emotet and non-emotet congestion, as opposed to just emotet congestion. Each illustration runs for a different amount of time. Usually it takes less time to observe congestion patterns, but it can take up to 4 hours to observe the injection while visiting money-lending websites. More than 400 pcap files were captured, including both emotet-related and benign congestion. This variation in congestion, collected in the.pcap files, is extremely useful for our proposed model's training and test results.

Attributes Extrication and Marking

The toolset (NetMate) processes the pcap files and then calculates functionality based on the flows. It is an open source method for calculating network flow statistics on Linux platforms. It produces attributes values for flows in the collected trace files at the end of processing. A flow is a series of packets (in computing) sent over a period of time from a source to destination sockets. A collection of mathematical attributes and related function values characterise each network congestion. A characteristic is a descriptive statistic that can be determined from one or more PPC in a flow. Sockets values (source-destination), with protocols omitted by further collection until the flows are created. To terminate a relationship via UDP flows, the timeout value of the flow is evaluated. In our research, researchers use two parameters to determine when TCP flows should be terminated. One is the usual relation tear down, and the other is the flow time out value. Regardless of which of these events occurs first, the connection is terminated based on the criterion. The flow timeout value of 620 seconds was selected because it corresponds to the architecture of the Real-time Congestion Flow. In the flow, researchers consider numerous PPC in each direction and a payload of greater than 1-byte. As a result, it removes failed flows and payload values. From the captured trace files, the following statistical attributes are derived:

The data link layer is not included in the packet length, and the IP layer length is used to measure packet (computing) lengths. Inter-arrival times are measured in microseconds. A total of 41 flow attributes are considered when constructing a classifier. Transmission is done in both directions (bidirectionally), and then the direction of the flow is determined by the first packet. After converting flow statistics with Netmate, a total of 412,714 distinct flows are collected, which can be used as an input vector for ML algorithms. As a result, every network stream is represented by a collection of statistical attribute values. To ensure that the constructed model (framework) is not reliant on unique IPs, the source and destination IPs are omitted from the attributes set. To prevent creating a biased paradigm, overfitting values (protocol details and port id) are omitted from the Attributes collection. Any categorization based on the above parameter value may result in the creation of a biased classifier that is incompatible with new emotet models.

The purpose is to create an Emotet CCS IP library file (2021). which is run by a different researcher and contains all the decrypted files (config) up to version 191. Researchers considered the details in the archive to be compatible with our defined conclusions, but it includes additional customization for models researchers hadn't seen before. As a result, researchers created a code snippet and a script (Python) to retrieve all encrypted files (ServConf and the config.conf) and remove the intermediary IPs. This operation is carried out for each Emotet botnet. Following the removal of duplicate IPs, an Emotet CCS IPs library file was developed, containing 3105 specific CCS IPs. This library was used in our class labelling operation.

Categorization Models (Training)

For a supervised categorization query, a categorization model would be built on earlier classified flow contexts. Researchers built a repository file for all intermediary CCS IPs from all Emotet releases and tested up to 191 in it. Each flow was named based on the results of a python script that compared the IPs in all streams with identified Emotet CCS IPs. As previously mentioned, memory strings could be used to extract CCS IPs for class labeling. Another alternative is to use the Emotet resource file

decoder to remove CCS IPs from each downloaded config.conf file. To retrieve IP values from the thread of memories or the text files, the emotet illustration should be run for each version in both directions. Extrication of IPs in this manner is a time-consuming and often contentious process. For example, researchers discovered that used IPs for down loading Emotet base files could be removed from thread memory during the execution of the study. As a result, certain IPs can be overlooked in this manner.

Researchers used the attributes set to evaluate the classifiers specified. The discovered classifier is then subjected to optimization of hyper-parameter and function identification in order to improve total TPR and accuracy. For our study, researchers used the open source Weka ML platform. In this section, researchers give a brief overview of each of the ML algorithms used in the research.

Random Forest (RF)

A RF is a group of decision trees [DT] used in an ensemble learning technique for categorization. Each tree associates and predicts when to codify newer data cases, and the final outcome is determined based on the collective voting of the trees. Consider a training illustration of five data instances [K1, K2, K3, K4, K5] with corresponding labels [J1, J2, J3, J4, J5]. RF will produce three separate DTs based on three divergent subsets of training instances: [K1, K2, K3], [K1, K3, K5], and [K2, K4, K5]. When classifying a new case, each RF tree makes a guess, and the final outcome is evaluated as the plurality of votes from its DT. RF performs well even in the face of “noise” since its final decision is based on plurality voting from its many DT (abnormal data). While noise can impact and affect a single DT, gathering the effects from several DT decreases the impact of disturbance (noise).

Multilayer Perceptron (MLP)

MLP is categorised as a type of ANN (Artificial Neural Network) based on BP (Back-Propagation) for distinguishing instance parameters. The MLP model’s matrices are modified during the process of training to decrease the error. Back propagation is used to make those weight and bias corrections proportional to the defect. The error can be calculated in many ways, by using the root mean squared error (RMSE).

Sequential Minimal Optimization (SMO)

The classifier built into the Weka method, which researchers use for categorization modeling, uses John Platt’s SMO algorithm for training the classifier SVM (Support Vector Machine). The SVM classifier’s training will result in a quadratic programming problem and can be used to solve quadratic programming problems. SMO scaling is done for different test problems because it avoids the matrix calculation, which is a costly computation. Since SVM evaluation dominates SMO computation,

Logistic Regression Model (LRM)

The logistic regression model, which is analogous to a sequential classifier in that it forecasts the target class using computed logits, belongs to the supervised categorization algorithm class (score). It calculates the correlation amongst descriptive statistics and operates logistic functions to approximate uncertainties. The Basic LC (Logistic classifier) created in Weka utilises Logit Boost with specific regression methods as core peers to accommodate the LRM. Simple Logistic regulates the optimum number of loops (iterations) for Logit Boost using cross-validation, resulting in automated attribute selection for increased categorization efficiency.

THROUGHPUT METRICS

In our categorization model, instances of classes Emotet (Positive), while non-Emotet (Negative). The values used to assess the accuracy of the classifiers is given in Fig.3

Figure 3. Parameters for Research Evaluation

$\Rightarrow TPR = \frac{TP}{TN + TP}$	$\Rightarrow FPR = \frac{FP}{TP + FN}$
$\Rightarrow precision = \frac{TP}{TP + FP}$	$\Rightarrow recall = \frac{TP}{TP + FN}$
$\Rightarrow F - Measure (F1 - Score) = 2 * \frac{precision * recall}{precision + recall}$	

- TP (True Positive) – Accurate categorization of Emotet instances.
- TN (True Negative) - Accurate Stratification of Non-Emotet Instances.
- FP (False Positive) – Incorrect Stratification of non-Emotet instances.
- FN (False Negative) -Incorrect Stratification of Emotet into defined class.
- FNR -False Negative Rate (Use to calculate Error Rate).
- Precision – use to measure Accuracy.
- Recall – Use to calculate Error Rate.
- FPR- True Positive Rate (use to measure Accuracy.)
- F-Measure –mean(Harmonic) value of recall and precision.

Results of Tested Classifiers

Overall precision is an important parameter to consider when using a ML technique to classify data. In certain cases, especially when dealing with intrusion detection and prevention, the dataset can be extremely unbalanced. The concerned class is a unique one, accounting for less than 11% of all groups. In order to minimise overall error, data mining algorithms typically focus on the large percentage class rather than the marginalised group. However, this technique could pose a significant problem in intrusion detection and prevention since inaccurate identification of the community of concern could cost far more than the plurality class.

In our dataset, a total of **481,639** non-emotet and **2143 emotet** stream (flow) associations were found. Despite the fact that the constructed classifier classifies all of the flows as non-emotet, researchers were able to achieve 99.83 percent accuracy. The integral classifier will miss any emotet streams, putting the machine at risk. As a result, this extremely unbalanced dataset poses a significant problem that the built-in classifier can effectively address. Subsampling may be used in similar situations to make the instance numbers for both groups identical. However, in NIDSs, which would deal with largely imbalanced datasets due to minority intrusions, this is not a good option. Subsampling causes an equivalent number of occasions to overfit the training and testing results, resulting in poor throughput in real-world scenarios. As a result, researchers used the data to train the classifier without any subsampling operations. As a result, rather than relying solely on total precision, researchers rely on the TP rate when evaluating our classifier.

Researchers created a code snippet script (Python) to mark the previously described Attributes gleaned from in-depth analysis of over 361 emotet illustrations Malign congestion was created by interacting with common Internet domains, while benign congestion was created by using real-world Emotet malware illustrations. During complex research, prominent weblinks in the Alexa top 600

sites are overlooked to populate by benign congestion. From all datasets, **481,639** not-Emotet and **2143** Emotet network flows were acquired by the end of the labelling process using established code snippets.

For our categorization problem, the authors used WEKA, an open-source platform for data mining activities. Having easy access to interfacing (GUI) with several supporting ML algorithms. To begin, the authors use some efficient classifiers such as RF, MLP, Naive Bayes, SMO, and LRM to construct the modelling using an approach (10-fold-cross) validation. At this stage, the authors want to find a reliable classifier for detecting emotet flows that doesn't need any hyper parameter optimization. All the function datasets are uniformly positioned into 10-uniform datasets in 10-fold cross validation. Each iteration uses 9 out of 10 values of the dataset for training purposes only, leaving the dataset for authorization. This procedure was replicated ten times to get validation, and the remaining nine data sets have been used for training. Finally, ten outcomes are combined to provide a single estimate.

The RF, MLP, SMO, and LRM classifiers were also evaluated for this category after the above attribute vectors were evaluated for each of the datasets. With TPR approach 1, and FPR approach 0, an appreciable outcome is acquired to get the accuracy of a verified classifier. First, the RF classifier is used to acquire the function data set vector. According to the findings of the checked classifier, **473,821** instances were accurately categorised, a rate of **99.9716** percent. There were **228** flow cases that were miscategorized in total. Table 2 shows the RF Classifier (Confusion Matrix Result Set). The model with the listed classifier takes **512.46** seconds to construct.

The findings show that eight non-emotet flows have been graded as emotet flows. Table 3 shows the computed efficiency metric for the constructed model. Both groups have a weighted mean TPR of **0.9989**.

Second, the authors used the MLP algorithm to build a classifier. This model takes **1824.46** seconds to create, which is slower than the RF output. 0.5 is the learning rate, 0.4 is the momentum, and 550 is the training time, the chosen classifier is run. According to the MLP results, **99.8213** percent of the flow instances are correctly categorised (**413,254** out of **527,591**). Table 4 shows the MLP Classifier (Confusion Matrix Result Set). The authors are seeing higher accuracy, but the TPR is not good at **0.613**. Our classifier misses nearly half of the emotet flows. According to the uncertainty matrix, **37** non-Emotet flows are graded as Emotet flows, indicating a high number of FP for non-Emotet flows. With the acquired confusion matrix, the output parametric values of the constructed framework can be conveniently evaluated. Table 5 shows the MLP Classifier (Throughput Parameter Results).

Third, for the classifier, the authors use the SMO algorithm with a tolerance parametric value of (0.0001) and an epsilon value of (1.0E-12). The model takes **3400.24** seconds to build. Aside from the model's build time efficiency, other throughput figures do not have the same values as the prior ones. There were 481,639 systematically categorised values (Iterations) and 2143 incorrectly categorised values (Iterations), for a 99.8391 percent categorization accuracy. Tables 6 and 7 show the SMO Classifier (Confusion matrix) and SMO Classifier (Throughput results) respectively.

Despite the high precision of the uncertainty matrix results, all emotet variables are listed as non-emotet flows (streams). Because of the imbalanced dataset, total precision is insufficient to demonstrate the classifier's output.

Finally, for the categorization problem at hand, the authors choose the LRM classifier. The LRM classifier takes **53.35** seconds to construct the categorization model. If the desire is to get a faster categorization outcome, this result is much superior. After designing and checking the model, the authors got **434,187** correctly categorised instances (**99.77** percent) and **1620** incorrectly categorised instances. The TP rate of the logistic classifier is **0.296**, which means that **1388** out of **2143** emotet flows are listed as non-emotet. Although the classifier's overall accuracy is still good, the TP rate is low.

Tables 8 and 9 demonstrate the LRM Classifier (confusion matrix) and LRM Classifier (throughput parameters) respectively.

The tested classifiers are compared graphically in Figures 4 and 5. As seen in Fig. 5, RF has the highest categorization accuracy of **99.9716** percent, while Singular Minimal Optimization has a lower categorization accuracy of **99.8391** percent, making it less efficient in verified classifiers. As seen in Fig. 4, the true positive rate graphic is plotted as another success criterion. These graphs also show that RF can detect emotet and classify flows effectively.

Table 2. RF Classifier (Confusion Matrix Result Set)

Non-Emotet Flow	Emotet Flow	Categorized as
413,689	11	Non-Emotet
223	1934	Emotet Flow

Table 3. Throughput results of RF Classifier.

	Precision	FPR	Recall	TPR	F -Measure	Class (Cluster)
Weight (Mean)	0.9989	0.1121	1	1	1	Non Emotet Flow
	0.9976	0	0.912	0.912	0.947	Emotet Flow
	0.9989	0.1121	0.9989	0.9989	0.9989	All

Table 4. MLP Classifier (Confusion matrix Result set)

Emotet Flow	Non-Emotet Flow	Categorized as
37	456,282	Non- Emotet Flow
973	981	Emotet Flow

Table 5. MLP Classifier (Throughput parameter results)

	Precision	FPR	Recall	TPR	F-Measure	Class (Cluster)
Weight (Mean)	0.9981	0.498	1	1	0.9989	Non-Emotet
	0.982	0	0.613	0.613	0.779	Emotet
	0.9981	0.57	0.9981	0.9981	0.9981	All

Table 6. SMO Classifier (Confusion matrix).

Emotet Flow	Non-Emotet Flow	Categorized as
0	481,639	Non-Emotet Flow
0	2143	Emotet Flow

Table 7. SMO Classifier(Throughput results).

	Recall	F-Measure	FPR	Precision	TPR	Class (Cluster)
Weight (Mean)	1	0.9981	1	0.9976	1	Non-Emotet
	0	0	0	0	1	Emotet
	0.9981	0.9943	0.9976	0.9989	0.9976	All

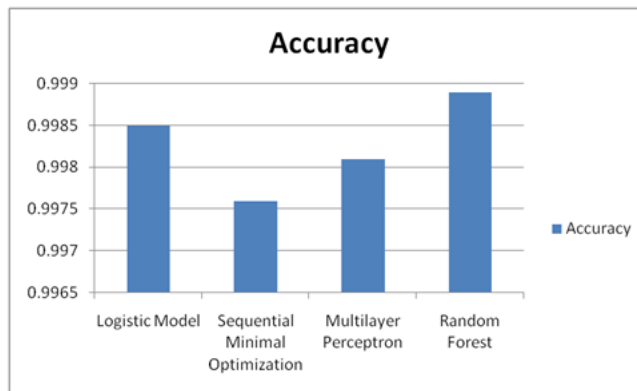
Table 8. LRM Classifier(Confusion matrix).

Non-Emotet Flow	Emotet Flow	Categorized as
464,182	216	Non-Emotet Flow
1388	512	Emotet Flow

Table 9. LRM Classifier (Throughput parameters)

	Precision	TPR	F-Measure value	FPR	Recall	Class (Cluster)
Weight (Mean)	0.9985	0.9989	0.997	0.786	0.9989	Non-Emotet
	0.723	0.296	0.413	0.002	0.296	Emotet
	0.9976	0.9985	0.9976	0.863	0.9985	All

Figure 4. Categorization accuracy of tested classifiers



Optimization of RF in Emotet Detection/Prevention

RF is an ensemble categorization and regression technique that outperforms all other data mining algorithms in terms of precision. Learning from an unbalanced dataset, as discussed in the previous section, is a significant challenge in ML and intelligence discovery applications. RF, on the other hand, performed admirably and outperformed the other students. It outperforms the Emotet dataset, which contains only 0.6 percent of all emotet flows, and has a higher category throughput. RF's bagging and random subspace mechanics demonstrate their effectiveness in lowering the correlation between each DT, which tends to reduce sources of error, including bias and variance.

In respect of noise computational analysis, the outputs of randomised unpruned DT will yield a stable categorization outcome. The RF categorization system is depicted in Fig. 6.

For two reasons, randomness is given for each DT in RF: To begin, the randomness is accomplished by building each DT with a separate bootstrap illustration (bootstrapping) from the overall dataset. As a result, there is less resemblance between the learners. After introducing newer

The classifier will make a final decision based on the DTs' plurality voting (bagging). Another benefit of the bagging mechanism is that it aids in assessing the learner's test success during the training phase. In the training process, nearly two-thirds of the designed dataset values are used in tree preparation, and the remaining one-third is used for testing purposes. Out of bagging (OOG) cases will therefore have greater categorization precision, preventing overfitting in the offline process.

The unpredictable collection of attributes in the attributes set to be used in separating each node at a DT adds to the randomness. The random subspace function would aid in the development of reasonably uncorrelated DT and would also reduce bag error. The similarity between the individual classifiers and the ensemble determines the RF learner's categorization accuracy. If all the trees in the forest were equal, each tree would make the same decision, resulting in a skewed prediction at the end.

The function selection process is one of the most difficult aspects of ML applications. Attribute selection necessitates a high level of experience and subject understanding in many areas. Using irrelevant functionality will

result in higher computing costs and an improvement in the classifier’s inaccuracy rate. With careful function selection, it is possible to avoid overfitting and arrive at a cost-effective standard. However, the benefits of function selection would add to the modelling task’s sophistication. The figure shows the categorization accuracy of tested classifiers.

Extraction of attributes from the network is a time-consuming and expensive operation. In addition, accounting for non-essential network attributes raises both the computational cost and the inaccuracy rates. Furthermore, since they neglect function construction, unrefined files (audit.pcap) are unsuitable for constructing intelligent vulnerability frameworks. As a result, the authors want to pick the attributes that are most important for categorization accuracy and computational cost reduction.

The Weka app is used for machine learning (analysis) to gather data attributes. For selecting an attribute set, the “Pick Attributes” panel offers many search methods. Some of the search strategies are: GA(Genetic-Algorithm), BF (Best-First), LFS (Linear-Forward-Selection), AR (Attribute-Ranking), and Forward Identification (Consistency Size-Subset). Correlation, continuity, and entropy-based approaches are examples of assessors. Different search and validation approaches may be merged to find the best function range for the problem at hand. Table 10 shows the effects of using a hybrid attributes assessor to test some of Weka’s attributes search methods for emotet intrusion detection and prevention. Figure 5 shows the average TPR of tested classifiers.

Figure 5. Averaged TPR of tested classifiers

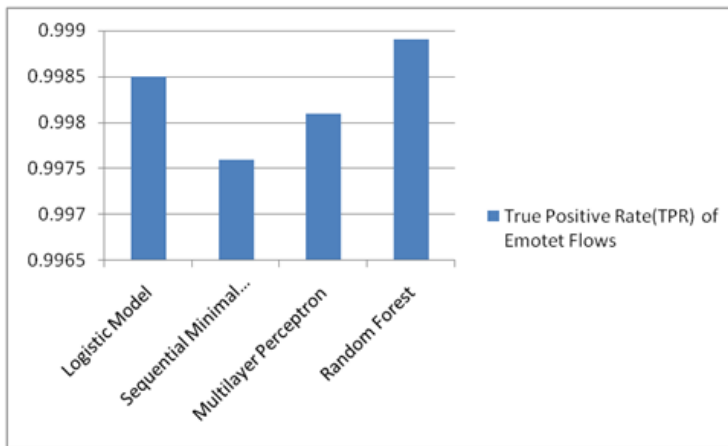


Figure 6. RF Classifier.

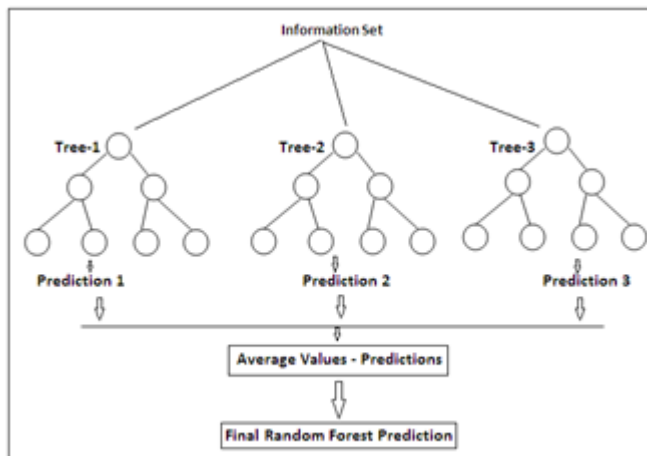


Table 10. Resemblance of SA(search algorithm) and Attributes Assessor for values selection.

SA(Search Algorithm) + Attributes Assessor	Authenticated Class. of Emotet / 2143	TPR of Emotet Flows	Overall Accuracy	OOG Error
1. There is no feature selection.	1921	0.9125	99.9216	0.0007
2. Assessing the Best First + Consistency subset.	1935	0.89812	99.9521	0.0008
3. Assessment of the Genetic Search + Consistency subset.	1891	0.87513	99.9397	0.0008
4. Consistency subset assessor + Linear forward catalogue	1998	0.89128	99.9294	0.0004
5. Chi Squared Attribute assessor + Ranker	1976	0.89468	99.9534	0.0008
6. Attribute Assessor + Ranker.	1937	0.89934	99.9592	0.0006
7. OneRAttribute _assessor + Ranker	1907	0.88931	99.9387	0.0005
8. Forward collection of subset size + Consistency Subset assessor	1911	0.92191	99.9726	0.0004

According to the findings, the Forward Identification (Consistency Size-Subset) search tool paired with a combined quality subset assessment methodology has the highest overall accuracy and TP levels. As a result, the authors refine our RF classifier using the attributes determined by this combination. Table 11 lists the characteristics that were chosen for this mix.

Table 11. Selected Attributes list by Forward Identification (Consistency Size-Subset) Assessor .

Selected Attributes s
Forward capacity Cumulative.
The total amount of backward capacity.
Prior to idle, the minimum time flow was operational.
The average frequency flow was involved prior to being idle.
Before being involved, the maximum time flow remained idle.
Maximum packet(Computing) size for forwarding.
Maximum size of a backward packet(Computing)
Forward capacity Cumulative.

Finally, the authors can optimise the number of DT (A), randomly chosen attributes (B), seed (C), and depth parameter (E) of each tree for emotet intrusion detection and prevention using RF hyper parameters. Optimizing the number of DT reduces uncertainty and, as a result, the likelihood of overfitting. On the other hand, increasing the number of classifiers would lengthen the time it takes to construct the model and classify it. For our emotet classifier, the authors discovered that a categorization tree number of 150 is the best choice.

Another important parameter (which manages unpredictability in the tree) for splitting nodes is randomly chosen attributes for each tree. This aids in the forest’s robustness and power. Related to Breiman’s research, using $\lceil \log_2(\text{Attributes_id}) + 1 \rceil$ Attributes_id in the ensemble tree construction is an appreciable idea. This extrication is supported by the acquired data. The best TP rate and categorization accuracy are achieved when the authors use 5 as the B parameter value. The depth

of each tree is determined by the tree depth parameter. When the value is equal to 1 of the parameters (depth), the tree depth is infinite. When compared to other options, it has been discovered that selecting an infinite tree depth yields better outcomes. A seed is a random number that is used to ensure that the same model is generated every time the same value is entered. To improve precision, the seed parameter should be set to 2. Table 12's third panel, which is outlined in purple, contains the best hyperparameter set combination. With these hyper parameter values, the authors improved overall accuracy and TPR.

Table 12. Selected Attributes list by Forward Identification (Consistency Size-Subset) Assessor.

Tree Number(A)	Attributes number (B)	Seed(C)	Tree depth (E)	TP Rate	Accuracy (%)	Time (Sec.)
30	5	2	1	0.903	99.9483	39.93
60	5	2	1	0.915	99.954	132.83
150	5	2	1	0.917	99.9555	244.15
300	5	2	1	0.915	99.954	497.19
350	5	2	1	0.914	99.9538	714.5
200	4	2	1	0.909	99.9516	199.86
150	6	2	1	0.916	99.954	306.15
150	7	2	8	0.692	99.838	342.21
150	7	2	16	0.816	99.9041	401.2
150	7	2	20	0.912	99.9516	392.71
150	9	2	1	0.915	99.9501	516.08
150	5	3	1	0.914	99.9534	252.06
150	5	4	1	0.913	99.9537	248.15
150	7	3	1	0.915	99.9531	770.14
350	7	3	1	0.914	99.9531	1212.26
350	7	4	1	0.917	99.9546	1188.12
350	9	4	1	0.912	99.9468	1518.11

COMPARISON WITH AVAILABLE TECHNIQUE

A limited number of research papers are available only focusing on Emotet Malware.

The proposed approach is better as it focuses on the best set of hyperparameters for emotet flow identification and emotet detection by congestion flows and generates results by evaluating packet statistical attributes. Below is the comparative study of research work found till date.

CONCLUSION AND FUTURE WORK

For more than two years, the Emotet money-lending Trojan class has posed a challenge to monetary institutions and their associated customers. It is becoming more risky with recent changes to its infection techniques and attack vectors. The authors created a classifier that can detect emotet infection by looking for emotet-related flows. For the categorization issue at hand, the authors compared the outcomes of four distinct classifiers. According to the analysis, the RF classifier outperforms the other four classifiers in terms of categorization accuracy. To improve the accuracy of the RF classifier, the

Table 13. Comparison with Available Technique

Reference	Technique	Focus
(Song et al., 2021)	For analysis Malware PowerShell scripts, feature optimization method is used.	RF (Random forest) model with (5-token), (3-grams) and the DL models with AST (abstract syntax tree) have detection rates (98%).
(Lallie et al., 2021)	Analysis of PowerShell script	Malicious file attached about COVID-19 infection status
(Azab & Khasawneh, 2020)	Malware Spectrogram Image Classification (MSIC)	Spectrogram images Analysis in malware file classification using CNN(convolution neural network)
(Kono et al., 2020)	Spam email Analysis for Malicious macros	Emotet malware family
(Lison & Mavroeidis, 2017)	Data-driven approach for detecting malware-generated domain names using RNN (recurrent neural networks).	Analysis of Malware Families
Proposed Work	Based on best set of hyper parameters for Emotet flow identification	Emotet Detection by congestion flows.

authors use the Forward Identification (Consistency Size-Subset) Assessor to refine the attribute set. The authors have run tests to find the best set of hyper parameters for emotet flow identification, which includes 150 categorization trees, four random attributes for tree splitting, one for depth, and two for seed. With a TPR of 92.3 percent, the authors achieved 99.9726 percent categorization accuracy. Emotet-related streams (flows) and malfunctioning can be quickly detected on networks using a defined and designed framework (Model).

To identify and dig out fingerprints of network malware (trojans) and to concentrate on certain up-to-date money-lending Trojans (Gozi (Ursnif)) and the Game Over Zeus (Zeus clone), poisoned across 600.000 systems across the globe. The authors will also keep an eye on Emotet to see if there are any new infection strategies or attack vectors in newer models.

For future work, the changing attributes and vector features of malware classes should be observed.

STATEMENT FOR CONFLICT OF INTEREST

There are no conflicts to declare.

DISCLOSURE OF POTENTIAL CONFLICTS OF INTEREST

None

INFORMED CONSENT STATEMENT

None

FUNDING

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

ACKNOWLEDGMENT

None

REFERENCES

- Azab, A., Layton, R., Alazab, M., & Oliver, J. (2014, November). Mining malware to detect variants. In *2014 fifth cybercrime and trustworthy computing conference* (pp. 44-53). IEEE. doi:10.1109/CTC.2014.11
- Azab, A., & Khasawneh, M. (2020). MSIC: Malware spectrogram image classification. *IEEE Access: Practical Innovations, Open Solutions*, 8, 102007–102021. doi:10.1109/ACCESS.2020.2999320
- Ceschin, F., Botacin, M., Gomes, H. M., Oliveira, L. S., & Grégio, A. (2019, November). Shallow security: On the creation of adversarial variants to evade machine learning-based malware detectors. In *Proceedings of the 3rd Reversing and Offensive-oriented Trends Symposium* (pp. 1-9). doi:10.1145/3375894.3375898
- Casino, F., Totosis, N., Apostolopoulos, T., Lykousas, N., & Patsakis, C. (2021). Analysis and Correlation of Visual Evidence in Campaigns of Malicious Office Documents. *arXiv preprint arXiv:2103.16143*.
- Daku, H., Zavorsky, P., & Malik, Y. (2018, August). Behavioral-based classification and identification of ransomware variants using machine learning. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)* (pp. 1560-1564). IEEE. doi:10.1109/TrustCom/BigDataSE.2018.00224
- Gezer, A., Warner, G., Wilson, C., & Shrestha, P. (2019). A flow-based approach for trickbot banking trojan detection. *Computers & Security*, 84, 179–192. doi:10.1016/j.cose.2019.03.013
- Grammatikakis, K. P., Koufos, I., Kolokotronis, N., Vassilakis, C., & Shiales, S. (2021, July). Understanding and Mitigating Banking Trojans: From Zeus to Emotet. In *2021 IEEE International Conference on Cyber Security and Resilience (CSR)* (pp. 121-128). IEEE. doi:10.1109/CSR51186.2021.9527960
- Kono, K., Phomkeona, S., & Okamura, K. (2018, July). An unknown malware detection using execution registry access. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 2, pp. 487-491). IEEE. doi:10.1109/COMPSAC.2018.10281
- Kono, K., Phomkeona, S., & Okamura, K. (2018, July). An unknown malware detection using execution registry access. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)* (Vol. 2, pp. 487-491). IEEE. doi:10.1109/COMPSAC.2018.10281
- Lallie, H. S., Shepherd, L. A., Nurse, J. R., Erola, A., Epiphaniou, G., Maple, C., & Bellekens, X. (2021). Cyber security in the age of covid-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic. *Computers & Security*, 105, 102248. doi:10.1016/j.cose.2021.102248
- Lison, P., & Mavroeidis, V. (2017). *Automatic detection of malware-generated domains with recurrent neural models*. arXiv preprint arXiv:1709.07102.
- Niu, W., Li, T., Zhang, X., Hu, T., Jiang, T., & Wu, H. (2019). Using XGBoost to discover infected hosts based on HTTP traffic. *Security and Communication Networks*, 2019, 2019. doi:10.1155/2019/2182615
- Rajawat, A. S., Rawat, R., Barhanpurkar, K., Shaw, R. N., & Ghosh, A. (2021). Vulnerability Analysis at Industrial Internet of Things Platform on Dark Web Network Using Computational Intelligence. *Computationally Intelligent Systems and their Applications*, 39-51.
- Rawat, R., Mahor, V., Chirgaiya, S., Shaw, R. N., & Ghosh, A. (2021). Analysis of Darknet Traffic for Criminal Activities Detection Using TF-IDF and Light Gradient Boosted Machine Learning Algorithm. In *Innovations in Electrical and Electronic Engineering* (pp. 671–681). Springer. doi:10.1007/978-981-16-0749-3_53
- Rawat, R., Mahor, V., Chirgaiya, S., & Rathore, A. S. (2021). 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
- Song, J., Kim, J., Choi, S., Kim, J., & Kim, I. (2021). Evaluations of AI-based malicious PowerShell detection with feature optimizations. *ETRI Journal*, 43(3), 549–560. doi:10.4218/etrij.2020-0215
- Soomro, T. R., & Hussain, M. (2019). Social Media-Related Cybercrimes and Techniques for Their Prevention. *Appl. Comput. Syst.*, 24(1), 9–17. doi:10.2478/acss-2019-0002

Romil Rawat attended several research programs and received research grants from USA, Germany, Italy, and UK. The author has research alignment towards Cyber Security, IoT, Dark Web Crime analysis and investigation techniques, and working towards tracing of illicit anonymous contents of cyber terrorism and criminal activities. He also chaired International Conferences and Hosted several research events including National and International Research Schools, PhD colloquium, Workshops, training programs. He also published several Research Patents. .

Sanjaya Kumar Sarangi is an adjunct Professor and Coordinator at Utkal University, distinguished background in Academic, Research, and Industry sectors combined with 23 years of experience in the knowledge towards achieving the institution's objectives through skill, knowledge, and nourishing the global educational system. He is a qualified GATE and holds a Master of Technology in Comp. Sc and Engg. and Ph.D. degree in Computer Science. He was the Research Fellow in Science and Technology, at UGC, Govt of India, and visiting Doctoral Fellow at the University of California, USA. His research findings are in Wireless Ad hoc and Sensor Networks, IDS, Mobile Communications, Cloud Computing, Geospatial Science and Remote Sensing, Climate Change, and Disaster prediction system. He has a number of publications in Journals and Conferences. He has authored many textbooks and book chapters. He has more than 30 National and International Patents. He is an active member and Life member in many associations and also Editor, Technical Program Committee member, and reviewer in reputed journals and conferences. He is a successful coordinator for the NMICT project conducted by IIT, Bombay, and Kharagpur. His Contributed Digital Resources dedicated to Govt. of Odisha, IGNOU, and CSTA, USA. Academic coordinator in administration to various components of the Internationalization of higher education and international profile-building of university and academic exchange agreements with international universities. Managing International Student and Scholar Services with visibility of the international community and promoting educational and cultural exchange programs as International Student Advisor. Taking care of Information and Communication Technology (ICT) to enhance and optimize the information and worldwide research that can lead to improved student learning and teaching methods. Also undertakes ICT-based support for the smooth functioning of e-learning systems for the university.

P. WILLIAM is working as an Assistant Professor, Department of Information Technology, Sanjivani College of Engineering, SPPU, Pune. He received his Bachelor of Engineering in Computer Science and Engineering from CSVTU, Bhilai in 2013 and Master of Technology in Computer Science and Engineering from CSVTU, Bhilai in 2017. He is currently pursuing his Ph. D. in the Department of Computer Science and Engineering from School of Engineering & Information Technology, MATS University, Raipur. He has published many papers in Scopus indexed journals and IEEE Conferences. His field of research includes Natural Language Processing, Machine learning, Soft Computing, Cyber Security and Cloud Computing. He has been associated with numerous Multi-National Companies including IBM, TCS etc. and Educational Groups. A focused and hardworking professional with experience in taking Corporate Trainings. He is a Life Member of Quality Circle Forum of India (QCFI) and member of various other professional bodies. william160891@gmail.com

Snehil Dahima has a MCA and P.Hd. in Computer Application, having 19 Years of Experience in Teaching and 9 Years of Experience in Research. Area of Interests are distributed computing , cloud computing and data mining.

Sonali Gupta can be contacted at Sonali.goyal@yahoo.com

K.Sakthidasan Sankaran is an Professor in the Department of Electronics and Communication Engineering at Hindustan Institute of Technology and Science, India. He received his B.E. degree from Anna University in 2005, M.Tech. Degree from SRM University in 2007 and Ph.D. Degree from Anna University in 2016. He is a Senior Member of IEEE for the past 10 years and member in various professional bodies. He is an active reviewer in Elsevier Journals and editorial board member in various international Journals. His research interests include Image Processing, Wireless Networks, Cloud Computing and Antenna Design. He has published more than 70 papers in Refereed Journals and International Conferences. He has also published three books to his credits.