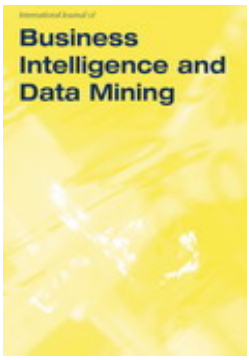



**International Journal of Business Intelligence and Data Mining**

ISSN online: 1743-8195 - ISSN print: 1743-8187
https://www.inderscience.com/ijbidm

# Identification of authorship and prevention of fraudulent transactions/cybercrime using efficient high performance machine learning techniques

B.J. Sowmya, R. Hanumantharaju, D. Pradeep Kumar, K.G. Srinivasa

**DOI:** 10.1504/IJBIDM.2022.10045310

**Article History:**
Received: 29 August 2021
Accepted: 13 December 2021
Published online: 30 November 2022

# Identification of authorship and prevention of fraudulent transactions/cybercrime using efficient high performance machine learning techniques

B.J. Sowmya*, R. Hanumantharaju and D. Pradeep Kumar

Department of Computer Science and Engineering,
M.S. Ramaiah Institute of Technology, India
and
Affiliated to: Visvesvaraya Technological University, India
Email: sowmyabj@msrit.edu
Email: hmrcs@msrit.edu
Email: pradeepkumard@msrit.edu
*Corresponding author

K.G. Srinivasa

Department of Information Management and Emerging Engineering,
National Institute of Technical Teachers Training and Research, India
Email: kgsrinivasa@gmail.com

**Abstract:** Cyber safety is the best skill required among a group of employees in organisations. Many offenders hide behind anonymous masking, including dishonest purchases, brazen plagiarism of the work, breaching corporate safety and stealing private data. To understand and analyse the actual phenomenon encountered with data, requirements of scientific methods, machine learning techniques, processes are to be used. This paper aims to tackle these problems by providing a protection layer for users, where data is being gathered from cyber security sources, analytical complement with latest data-driven patterns provides effective security solutions. We then discuss machine learning, deep learning powered models for the detection of insider threats and identifying authorship identification of anonymised articles. The individual modules are trained on authorship attribution, mouse monitoring, keyboard monitoring and command tracing and reached promising results with good accuracies in the range of 65%–85% on average.

**Keywords:** cybercrime; authorship attribution; machine learning; authorship identification; deep learning.

**Reference** to this paper should be made as follows: Sowmya, B.J., Hanumantharaju, R., Kumar, D.P. and Srinivasa, K.G. (2023) 'Identification of authorship and prevention of fraudulent transactions/cybercrime using efficient high performance machine learning techniques', *Int. J. Business Intelligence and Data Mining*, Vol. 22, Nos. 1/2, pp.144–169.

**Biographical notes:** B.J. Sowmya is currently working as an Assistant Professor in the Department of Computer Science and Engineering at M.S. Ramaiah Institute of Technology, Bangalore. Her area of interest is data

analytics, machine learning and internet of things. She is a co-author of several papers published in international journals, conferences, book chapters and a book.

R. Hanumantharaju is currently working as an Assistant Professor in the Department of Computer Science and Engineering at M.S. Ramaiah Institute of Technology, Bangalore. He is also a research scholar in the Department of Computer Science and Engineering at Siddaganga Institute of Technology, affiliated to Visvesvaraya Technological University. His area of interest is internet of things, edge computing, embedded systems and distributed systems. He is a co-author of several papers published in international journals, conferences, book chapters and a book on internet of things.

D. Pradeep Kumar is currently working as an Assistant Professor in the Department of Computer Science and Engineering at M.S. Ramaiah Institute of Technology, Bangalore. His area of interest is data analytics, machine learning and big data. He is a co-author of several papers published in international journals, conferences and book chapters.

K.G. Srinivasa is working as a Professor in the Department of Information Management and Emerging Engineering at National Institute of Technical Teachers Training and Research, Chandigarh (MHRD, Government of India). He is a recipient of All India Council for Technical Education – Career Award for Young Teachers, Indian Society of Technical Education – ISGITS National Award for Best Research Work Done by Young Teachers, Institution of Engineers (India) – IEI Young Engineer Award in Computer Engineering, Rajarambapu Patil National Award for Promising Engineering Teacher Award from ISTE – 2012, IMS Singapore – Visiting Scientist Fellowship Award. He has published more than hundred research papers in international conferences and journals.

## 1 Introduction

Everyone assumes that cybercrime is robbing anybody's personal details. However, one would certainly assume that cybercrime is about the use of an electronic system to steal or attempt to damage somebody's data on a network. It is also an unauthorised practice covering a range of concerns ranging from fraud to the use of the system or destination IP as a criminal weapon. Global terrorism has spread far past 10–20 years ago. Yet, cyber terrorism is not only linked to extremist groups or terrorists. However, cyber terrorism is just a means of intimidating someone or property to generate uncertainty. A vast population of billions and trillions of websites or users in this online world or cyberspace they also access it to shopping, entertainment, computer games, payments, e-commerce, etc. for several different reasons. Due to this strong rate of growth in the last decade, everybody can easily hit everything in this age of technology and high-speed internet access. Furthermore, the internet has created an information universe that can be accessible to anyone. This has contributed to a major rise in the crime rate, particularly cybercrime. In addition, owing to the improved connection speeds, the data circulation rate has also increased considerably. An analytical model's basic approach to fraud detection is to discover possible fraud predictors related to known fraudsters and their prior activities. The most powerful fraud models are founded on

historical data, just like the most powerful consumer response models. Both supervised and unsupervised techniques can be used for authorship identification.

Given the difficulty of monitoring human behaviour to detect fraud, some approaches along this line have been proposed to address some of the challenges. For example, some research tried to improve data processing precision and speed by using a hybrid automated learning system or incremental learning.

Another difficulty in fraud detection is a shortage of data from which detection systems may learn, presented a fraud-detection system that does not require previous fraudulent cases. In any event, fraud detection is a multifaceted problem in terms of human behaviour.

The main objective of this paper is to automate authorship identification of an anonymous articles, use authorship identification based on different attributes like usage profiling (using mouse/keyboard activities, system logs) and prevent internal security breaches, prevention of fraud and plagiarism attacks within organisations, utilise authorship attributions for helping victims of cyber bullying by revealing identity of cyber criminals and creation of a web application to make the above objectives possible and easily accessible.

This paper aims to identify authorship of a given anonymous article and verify the authorship of any article to prevent plagiarism. It also provides a graphical user interface application which monitors the resources of an organisation to prevent an insider attack, i.e., the application monitors the resources and its utilisation and uses a machine learning model to predict when an intruder is using the system, so that it might be brought into the notice of the organisation.

## 2 Literature survey

A systematic literature survey was undertaken to establish strategies for authoring. These approaches are based not only on the English language but also on many other languages including Arabic, German, Bengali, etc. Several reports have also been studied on intruder identification strategies. These methods have been listed below.

Alhijawi et al. (2018) have explained that identifying the authorship is the method by which the author of an unpublished text may be determined from the lists of possible authors accused of having written it. Thus, plagiarism, cyber bullying, and spamming can be observed quite usefully. The authors made an analysis of the different methods for attribution of authorship and they have used text dataset. The dataset was initially pre-processed for lexical characteristics, syntax characteristics and semantic characteristics. The functions were evaluated with appropriate machine learning algorithms following pre-processing. 70% of researchers used supervised learning for classification. Support vector machines (SVMs), nearest neighbours, decision trees, random forest and naive Bayes (NB) were among the most widely used algorithms. Unsupervised learning, in other words, these researchers has used clustering as a strategy. It was extremely capital demanding because only 4% of researchers used deep learning and in applications such as e-mail authorship, gender identity, source code authorship, and instant messaging authorship, this authorship identification technique is used. Bozkurt et al. (2017) have explored how a paper by a community of possible writers may be categorised. The framework of tf-idf with supporting vector machines, parametric methods such as Gaussian classifier and non-parametric methods like the Parzen

windows and nearest neighbours with various supervised and unsupervised methods of learning have all been done for this. In addition, attribute extraction methods have been introduced and essential aspects such as stylometry, variety of vocabulary, word bag and word function frequency have been extracted. The use of SVMs with a bag of terms and a Gaussian word set function was accomplished by a high degree of precision. Both these classifiers have been experimented and there was also a mixture of classifiers. Gaussian stylometry functionality set was 60% accurate and SVMs produced 95% performance in the word bag set. Although SVM's accuracy has resulted in higher datasets being computer consuming.

Dauber et al. (2017) have explained how stylometric features can be used in collective text documents and to correctly classify the authorship of such works. Stylometry used linguistic features of the text to derive written form from it to classify the document's author. The used algorithms included SVM and multi-labelling approaches. Wikia also compiled records used to guarantee multi-authored text. The attribution of authorship may play a large part in disclosing the identity of anonymously written texts, especially in recent times through the internet. The coordination serves as a medium to cover the normal stylometric characteristics and for such documents standard identification strategies fail. Therefore, experiments were conducted in these situations to assess the efficiency of different techniques. Both single and multi-authored established authorship records are considered of which knowledge of the number of researchers known in any text has been inferred. Collaborative, non-collaborative and pre-segmented text record creation was undertaken. A recognised community of authors A1, A2, …, An, describes the systematic issue of single authority attribution and one must detect the author of an anonymous text, each with its collection of documents d. For single author feature sets and methods research projects which demonstrated greater precision and extension for multi-authored documents were focused on previously conducted research. Write prints limited was used as a function set for multi-author papers. The top (m) author of a document generated by m authors with n probable authors was considered for evaluation purposes of the linear SVM. Within the developed experimental design, five-fold cross-validation was carried out to verify the robustness of stylometry techniques, thereby allowing a basis for comparing the results. A total of 60 writers were working, each followed by nine training documents. The SVM classifier used a threshold of 0.5 for the multi-label classifiers. Kallimani et al. (2019) have suggested a new method by using a stylometric approach to classify authors of publication. Authors gathered texts produced and developed a SVM model and helped to determine a new text document, by 50 separate writers. In addition, the study is carried out in Hindi, the regional language, which demands that it carry out a particular form of pre-processing, for example, eliminating specific punctuations in Hindi. The authors have used bi-gram, tri-gram, and n-gram to preserve the grammatical essence of the word. Comparative research on bi-grams, tri-grams and n-grams is being carried out between NB and the SVM model.

Swain et al. (2017) have discussed 46 research records and three publications concerned with the identification of authorships attribution (AA). The survey contains a variety of Arabic, German, English, Bengali, Latin, Persian and other languages. The knowledge comprises a wide range of subject areas including sports, news, literature, rhymes, poems, video, travel and tourism, text messages, e-mail spam, etc. The authors also address many ways of attributing the speaker. The author of a given source code, for example, may be identified. So, with the same author, several more malicious codes may

have been written, and this can be known. Authorship research of ancient Arabic texts yielded strong results with an accuracy of about 90%. Another research indicated that AA challenge and NLP-based characteristics were more consistent than BOW-based characteristics when presenting details on the words and verbs used. AA was performed in a collection of Arabic poetry, which gave 96% accuracy to classify their authors. The numerous AA forms and their applications were therefore explored (Hurtado et al., 2014) have explained how neural networks train a stylometric model. The writing style was based on the characteristics classified in four classes namely lexical, styling, syntactic and idiosyncratic characteristics. The algorithm used was a supervised learning model such as a regular multi-scale perceptron, a random forest with 12 random trees, supported vectors with polynomial kernel and neighbours with a neighbouring count of 10. After all features were integrated, the best results were obtained with validation methods such as ROC and area under the ROC curve (AUC). Identification of the intra-domain and inter-domain researchers was carried out. The domain or scope of the paper proved meaningless when finding the speaker, as the process of studying for both was very challenging. In addition, the number and number of features of the authors depended. 138 features were deemed ideal. It was also noticed that the model output decreased with the growing number of authors. These patterns were consistent for recognition of both intra-domain and inter-domain authors. The study also reported that the MLP with six characteristics yielded better results than the subset of characteristics.

Nirkhi et al. (2015) have focused primarily on the discovery of publishers of cybercrimes and identified the maliciousness of the mail or post. The misuse of the sender's address is usually used in cybercrimes such as identity fraud. Cybercrimes can be traced using the words printed by a cybercriminal using authorship recognition. The detection of authorship typically comes from two processes, the mathematical approach that involves MDS and the cluster analysis and machine learning techniques such as SVMs and NB. The support of word frequency and n-grams vector machines has been used and two datasets used to conduct analysis. The precision of the C50 dataset was 88% and of the Enron dataset 80% when 50 authors were considered, and the n-gram number used in both experiments was one. The result was that where there are multiple researchers and when the text is brief, the n-gram approach was better adapted. Dugar et al. (2019) have compared the conventional machine learning methods, which uses deep neural networks (DNNs) for authorship identification. Zhi Lu, a component package of Reuters Corpus Volume 1 (RCV1) was used to train the neural network with Zhi Lu's Reuters 50 50 dataset. The difficulty with neural networks is that the data must be fully numerical and not text-based. Thus, the data were first implemented by using Doc2Vec embedding, an unmonitored learning algorithm to learn the vector representation of the text with a variable length feature. The data was first transmitted through a neural network after convergence, resulting in an accuracy of 58% and this was improved by tuning hyperparameters. The activation function (RelU, sigmoid, tanh, etc.) was the hyper parameter chosen for tuning in the SoftMax function. The precision reached with rising the number of epochs is 67.6%, considerably higher than the accuracy without hyperparameter tuning. The paper thus provides a means of defining the authorship using DNNs and illustrates the relevance of tuning for hyper parameters.

Salem et al. (2008) have conducted surveys to achieve insider intrusion advancement especially in connection with the cyber and computer systems. A strong distinction among the two most critical insider assaults, i.e., attacks were setup by traitor and masquerade. Understanding that internal attacks with worm attacks were subsumed by

59% of the recorded cases of malicious insider attacks leading to big losses, the inner facets of defence need urgently to be tackled. An entity with genuine credentials and access is identified by a traitor, but due to shifts in intent of a malicious nature these credentials are misused against the safety instructions. On the other hand, masquerades capture a valid user's identity, so that access cannot be obtained. As masqueraders appear to have a small understanding of the mechanism, their behaviour, therefore, may be a helpful way to detect such attacks. They display significant improvements. But traitor attacks cannot be identified with profiling quickly. It was known that insider attacks would form part of one of many practices, such as package sniffing, malicious programmed installation and illicit operation utilising available resources. However, log files may also be used to classify certain events as they leave a trail behind. Organisations would track the identification mechanisms so no documents and traces can be detected without efficient tracking. The most internal risks occurred at the level of the programmed and not at the level of the network, and host-based surveillance were thus fundamental. However, as network sensors are easier to install, people usually spend, but most attacks will not even go up to a level. The accuracy of reported data relies heavily on internal threat identification. Calls or device calls are successful user activity identifiers to be tracked. Unix Shell commands could be modelled with a Markov chain. It was known that any tools in Windows and the network environment for profiling are accessible from the logs available. Software profiling has also been investigated and a distinct database must be prepared for each privileged operation.

Harilal et al. (2018) have analysed the TWOS: the wolf of SUTD dataset, which was created through a gaming contest which enabled users to be both natural and malicious. This work opens the analysis resources in this dataset. An attempt has been made to gather regular attack data and traitor data. Because of defective data like WUIL and RUU datasets, new data has been developed. A total of 320 logs spanning 24 users have been created. The rivalry was arranged to encourage six teams to participate in separate sales teams, to contact a similar group of customers and to try to accomplish as much. The competition was five days long (Mon–Fri), with teams required to be average and to lift their points on the first day of the competition. On the second day, teams were told who to target using the masquerade technique on a 90-minute notice (wildcard period). The third day involved dismissal and recruiting times to facilitate traitorous attacks and the fourth day was again for masquerades. The last day was when the points were registered and boosted if there was any chance. The data is obtained using Amazon EC2 servers and instances of Amazon workspace. For file servers, network proxy, etc., EC2 servers were used. For both systems, Windows domain controller servers were used, as they are prominently found in business environments. In addition, the log machine calls, and keyboard and mouse operation were generated by agents. There was also a host control agent. When the keyword is pressed/released and key type anonymised using zone categorisation, the keystroke data collection contains. Mouse activity has also been traced through press/release coordinates and mouse action. The control records file logs and generated and destroyed operations. Anonymisation has been prioritised in all documents. SMTP logs were used to monitor e-mail bodies.

Oladimeji et al. (2019) have conducted a comprehensive survey on current insider-threat identification methods. Any 87% of compliance failures were vulnerable to insider attacks in 70% of organisations. Machine-learning and non-machine-learning are the two main streams followed. The technique used often is a block chain for the

intrusion detection method, between non-machine learning techniques (IDS). This strategy tackles problems related to IDS: poor handling power overhead flow, small coverage of signatures and inspection. High levels of false positives are a big downside. A major limit for the identification of deviations is high false rates. A system like the Coburg utilities framework (CUF) is another major technique. It uses flow-based streaming electronic information, switches, or firewalls. Important approaches include the clustering, classification and use of the DNN among machine learning techniques. K-nearest neighbour (KNN) – the supervised learning algorithm used to automatically assign intrusion sensitivity values to the collaborative intrusion detection networks (CIDNs), a supervised learning algorithm. There are only a handful of IDS in RDBMS for protection. For each position, a typical behaviour profile is established, and anomalies are then identified. The unsupervised learning technology deep belief network (DBN) is used to learn insiders' behaviour and thereby track insiders' risks. This model uses four phases: log collection, log pre-processing, deep learning, insider learning and log classification. Elmasry et al. (2018) have focused on predictive identification of masquerades. High precision and low false alarms are the greatest problem. The authors introduced three profound learning models, namely DNNs, long short-term recurrent memory neural networks (LSTM-RNNs) and convolutional neural networks (CNNs). For this function, a UNIX command-based dataset is used. A static approach is introduced using DNN or LSTM-RNN models that are applied to static numeric data configurations and a dynamic approach is applied using a dynamically extracted CNN model from the user's text files.

Chen et al. (2020) introduced a novel framework for calculating semantic similarity: Siamese attention structure model deep reinforcement learning (DRSASM). Through reinforcement learning, the model automatically learns word segmentation and word distillation. To improve semantics, the entire design uses an LSTM network to extract semantic characteristics, followed by a novel attention mechanism model. The experiment shows that when compared to current base line structure models, this novel model using the SNLI dataset and Chinese business dataset can enhance accuracy. Yan et al. (2020) offers a dynamic partitioning technique based on a differential privacy mechanism. The geographical redundancy of location big data was decreased by adaptive density meshing and uniformity heuristic quad tree partitioning, and the temporal redundancy between adjacent data snapshots was minimised by sampling and differential processing of dynamic location big data. By modifying partition structures of the current dataset on the spatial structure of the preceding instant and adding Laplace noise, differential privacy protection has been achieved. Experiments using a cloud computing platform and real-world location big datasets show that the proposed algorithm can meet the dynamic partition release requirements of real-time location big data, and that the query precision of single-released location big data is better than other similar methods (Basodi et al., 2020). To identify malicious and secure measurements, develop a distance measure to be used as the cost function in deep-learning models based on feed-forward neural network architectures. These models are compared to existing state-of-the-art detection algorithms and supervised machine learning models in terms of efficiency and performance. The analysis shows better performance for deep learning models in detecting centralised data attacks.

Raghavan and El Gayar (2019) offers KNN, random forest, and SVM, as well as deep learning approaches such as autoencoders, CNNs, restricted Boltzmann machine (RBM) and DBN. The European (EU) dataset, as well as the Australian and German datasets,

will be utilised. The three assessment measures that would be employed are the AUC, Matthews correlation coefficient (MCC) and cost of failure. Shaukat et al. (2020) examines some of the most extensively utilised machine learning algorithms for detecting some of cyberspace's most dangerous cyber threats. DBNs, decision trees, and SVMs are the three basic machine learning algorithms researched. Based on commonly used and benchmark datasets, we offered a brief analysis of the performance of different machine learning algorithms in spam detection, intrusion detection and malware detection (Sánchez-Aguayo et al., 2021). This paper attempts to discuss current fraud detection research that incorporates the fraud triangle as well as machine learning and deep learning methodologies. We analysed research works relevant to fraud detection during the previous decade using the Kitchenham approach. This analysis shows that fraud is a hot topic under examination. Several studies on fraud detection using machine learning approaches were discovered, but there was no indication that they used the fraud triangle as a strategy for more efficient analysis.

In all the above related work, different neural network algorithms such as CNN, LSTM, SMTP logs were used which gives lesser accuracy than the proposed system. In the proposed system, the individual modules are trained on authorship attribution, mouse monitoring, keyboard monitoring and command tracing and reached promising results with good accuracies in the range of 65%–85% on an average. Hence, it is clear that the humongous threat of insider attack and cyber security in a company can be resolved with the aid of state-of-the-art machine learning techniques. One can train SVM and NB classifiers for Hindi language which also shows that machine learning and language processing are equally suitable for regional languages and these solutions can be deployed.

## 3 Methodology

The design of a platform for identifying authorship and preventing insider attack, fraudulent transactions and cybercrime using efficient machine learning and deep learning and high-performance computing techniques have implemented all the modules which will be integrated into one platform. The cost of insider threats (related to credential theft) for organisations in 2020 is $2.79 million. This alarming statistic is an indication of the importance of this application. The architecture of modules wherein each module is independent resulting in high performance and easy maintenance has been illustrated below. Modularity also makes the program more dynamic and decreases the latency by enabling multi thread execution of the modules. All these modules will give their individual classification, which will then be ensemble to give a unified characterisation based on which further decision making can be performed.

Figure 1 outlines the system architecture that is used for behaviour profiling and authorship identification, the dataset collection parameters are expressed as below:

- *KS:* Represents the keystrokes, collected from the keyboard data, which will be passed through an SVM classifier.
- *MT:* Represents the mouse traces, collected from the mouse movement, which will be passed through a decision tree classifier.

- *HM:* Represents the host monitor, collected from monitoring every system, which will be passed through a KNN classifier.
- *SL:* Represents the system logs, collected from the logged data of processes, etc., which will be passed through an SVM classifier.

**Figure 1** Architecture design (see online version for colours)

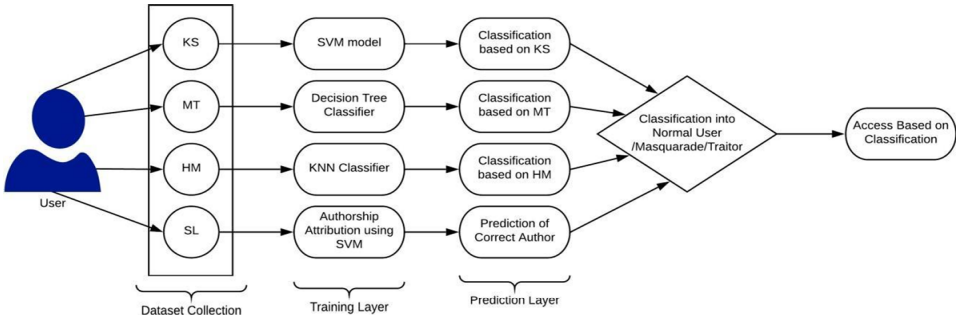



**Figure 2** UI design (see online version for colours)

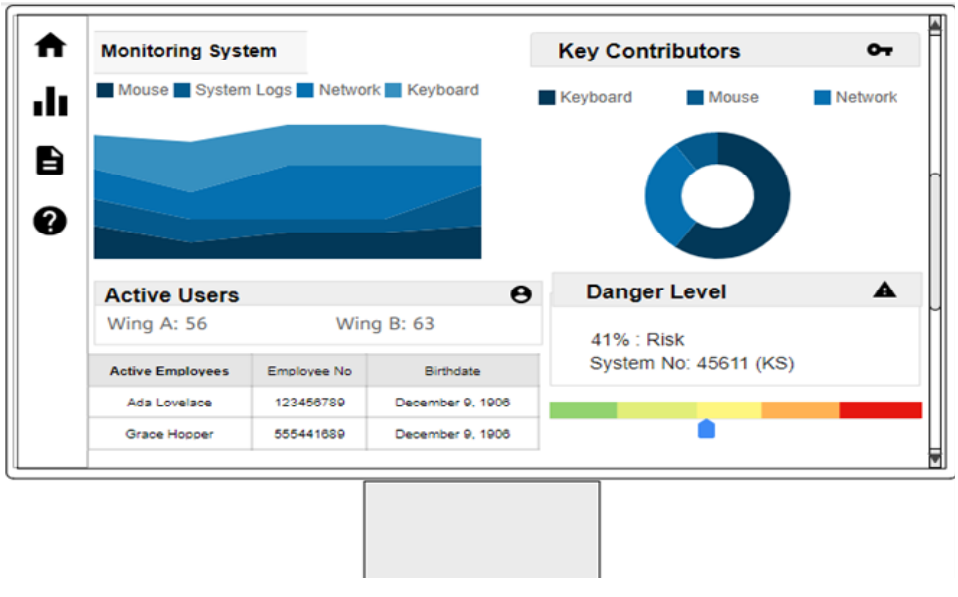



Figure 2 is a mockup of the user interface that was envisioned to create fulfilling the requirement of visualising the ongoing activity being recorded and analysing the threat level for internal security.

The first major decision will be to pick the type of service that must be fulfilled. The decision can be between the path of lightweight fast verification of any document for textual identification of the author and deciding to go with the loggers in case real-time monitoring is to be used. With the loggers incoming data and previously trained models of keystrokes, mouse tracking and host monitor one can classify the activity occurring and generate the threat level of the infrastructure being monitored.

## 4 Implementation

A modular approach has been followed for implementation. These modules include:

- data collection and pre-processing module
- feature extraction module
- training module
- prediction module.

This approach has been used for all the four major factors that determine any anomalous activity namely authorship attribution, command trace, keyboard trace and mouse trace. For implementing these modules, python is used as the programming language. The collected data from various sources which were collected by creating a real-time situation of a malicious attack. For classification machine learning algorithms such as NB and SVM are used which have shown promising results.

For development of graphical user interface, a desktop application which tracks the user activity in real-time, runs the machine-learning model in background and provides an output with an anomaly score graph on the output screen is developed. To implement this, Kivy framework is used, which is a python-based desktop application development platform.

**Figure 3** Algorithm description

```
Input: Data from different peripherals
begin:
  data collected from different peripherals are pre-processed
  results = []
  for every peripheral in peripherals:
    model_peripheral.fit(peripheral)
    result = model_peripheral.predict(peripheral)
    results.append(result)
  activity_level = ensemble(results)
end;
Output: Security Status of the system
```

As shown in Figure 3 the input from different peripherals namely keyboard, mouse, commands, and the text entered by the user. Now, this collected data is used for training machine learning models. Each module has its own model for prediction. All the prediction results are collected in an empty list result. This list is used to ensemble all the results and gives an output based on the combined result. As output, security status or safety level of the system is provided.

The modules for the complete implementation were divided as follows.

### *4.1 Authorship attribution module*

Authorship identification pertains to discovering the author of unknown documents. Whenever there is an incident of insider attack, when considering organisational security or cases like cyber bullying authorship attribution can play a major role in identifying the

perpetrator and said digital forensics. For the implementation of the authorship module, dataset and models are used:

- *Dataset:* The two datasets one in Hindi and English languages each, keeping in mind the importance of regional languages and the limited research work done in the same. The Hindi dataset comprises documents written by four different authors and a total of 2,089 files were used in the final training. For the English dataset training was performed on a total of 50 authors and documents authored by them. For each of the authors, at least 20 different documents were used for training. It must be understood here that to successfully implement authorship attribution a previously labelled dataset from each of the authors being tested must be available on which the model will be trained.
- *Data pre-processing and feature extraction:* For the Hindi model, the pre-processing included pruning out of punctuation marks like the Hindi danda and quotation marks. Then, the entire document was split into multiple files each containing 500 words and stored in a folder structure maintained with respect to the authors. Following this, the data is converted into vectors of numbers using bi-grams and tri-grams. Once the n-grams are generated the top 3,000 most frequently occurring bags are considered. For each of the 2,089 documents if that bag is a part 1 is entered for that column otherwise 0. Hence, the input matrix of features has the dimensions of (2,089 × 3,000). The output vector is of dimension of (2,089 × 1) representing true value for the correct author and false otherwise. For the English model on authorship attribution, the pre-processing involved removal of punctuation marks. A general practice of removing stop-words is followed when training on natural language text, but authorship attribution depends on the stylistic features which may be hidden in the stop words as well and hence they are not eliminated. The feature extraction is performed in a different manner, i.e., by using tf-idf vectors, converting the text into numbers.
- *Training:* For the training, two classification models are most used in machine learning, which are SVM and NB.
    a *Using SVM:* For classification of n-dimensional data, SVM is a very useful algorithm. For the 3,000 most commonly occurring bi-grams/tri-grams, weight is represented by theta. In SVM, the squared sum of weights is the objective function which must be minimised, after which when weight and input attribute are multiplied, we get a value greater than 1 or less than 1, which allows for classification.
    b *Using NB:* NB uses the Bayes probability theorem, where the predictors are assumed to be independent amongst themselves. Using the Bayes theorem, posterior probability is calculated, and the classification is achieved. For the English dataset due to the extensive nature of the dataset a tweaked version of NB, i.e., multinomial NB is used allowing for multiple classes.

### *4.2 Command trace module*

Command trace is essentially monitoring all the commands issued by the user to the system. This module traces all the commands and saves it into a text file. These monitored commands will then be used for training the machine learning model which

will predict any deviation from generic behaviour pattern or any malicious behaviour taking place. For the implementation of command trace module, following dataset and model is being used:

- *Dataset:* We have used a dataset with seeded masquerading users to compare various intrusion detection attacks. The data consists of 50 files corresponding to one user each. Each file contains 15,000 commands. The first 5,000 commands for each user do not contain any masqueraders and are intended as training data. The next 10,000 commands can be thought of as 100 blocks of 100 commands each. They are seeded with masquerading users, i.e., with data of another user not among the 50 users. At any given block after the initial 5,000 commands, a masquerade starts with a probability of 1%. If the previous block was a masquerade, the next block will also be a masquerade with a probability of 80%. About 5% of the test data contain masquerades.
- *Data pre-processing and model training:* Each of the 50 files is divided into 100 rows and 50 columns. One hundred rows are extracted from the commands from command number 5,000 to 15,000. These 10,000 commands are divided into the set of 100 commands resulting in 100 columns from each file. Fifty rows correspond to each user whose commands have been saved for analysis. If a particular set of commands, i.e., set of commands belonging to that row, is present for a particular author then his corresponding row will be marked 1, otherwise 0. This feature extraction results in an input matrix X of shape (5,000, 50), and output vector of shape (5,000, 1).
- *Training:* For training a machine learning model one has KNN since the goal is to classify data into two categories of malicious or non-malicious user. Therefore, a KNN classifier from sklearn library is imported. The model is divided into testing and training subsets respectively and trains our model on a training dataset.

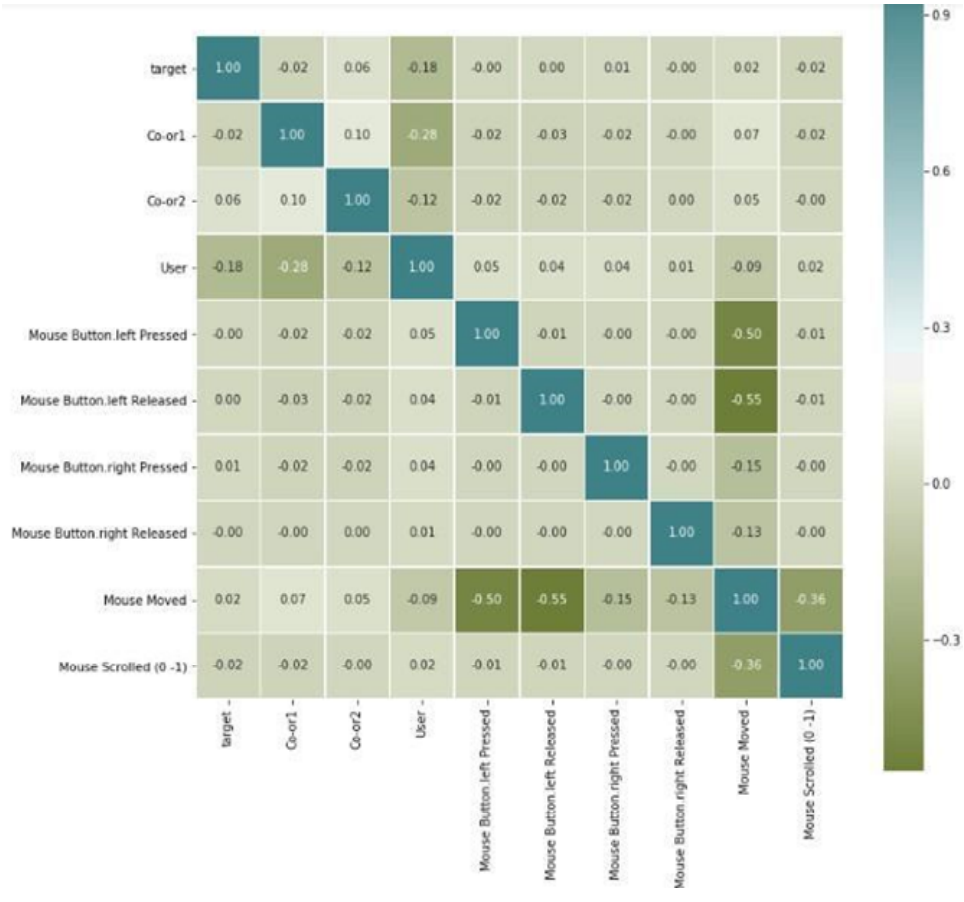
### *4.3 Mouse prediction module*

The mouse prediction module refers to the mouse movements made by the user. All the mouse movements during usage are recorded and saved to a text file. Through the data collected in the text file, a machine learning model is trained which predicts if any malicious behaviour/activity is taking place. For the implementation of mouse prediction module, the following dataset and model is being used:

- *Dataset:* The dataset being used consists of a log of the different operations carried out using a mouse. The total number of logs in the dataset is over 1.4 million. The attributes of the dataset are timestamp of the mouse activity, type of activity (mouse pressed, mouse moved, etc.), coordinates of the mouse, the user logged in, and finally the target attribute which tells if the mouse activity is normal, masquerade or a traitor.
- *Data pre-processing and EDA:* Before training any model, the dataset must be pre-processed according to the model being trained. For pre-processing, all the categorical attributes are first converted into numerical attributes. This is done through label encoding and one hot encoding. Label encoding was performed on the attributes user and target, and one hot encoding was done on the attribute direction.

After this, 5,000 entries were sampled from the dataset, for faster training time. The correlation matrix gives an idea of how the different attributes are correlated with each other providing valuable insight of the dataset. The correlation matrix of the dataset was found, and the correlation plot is as shown in Figure 4.

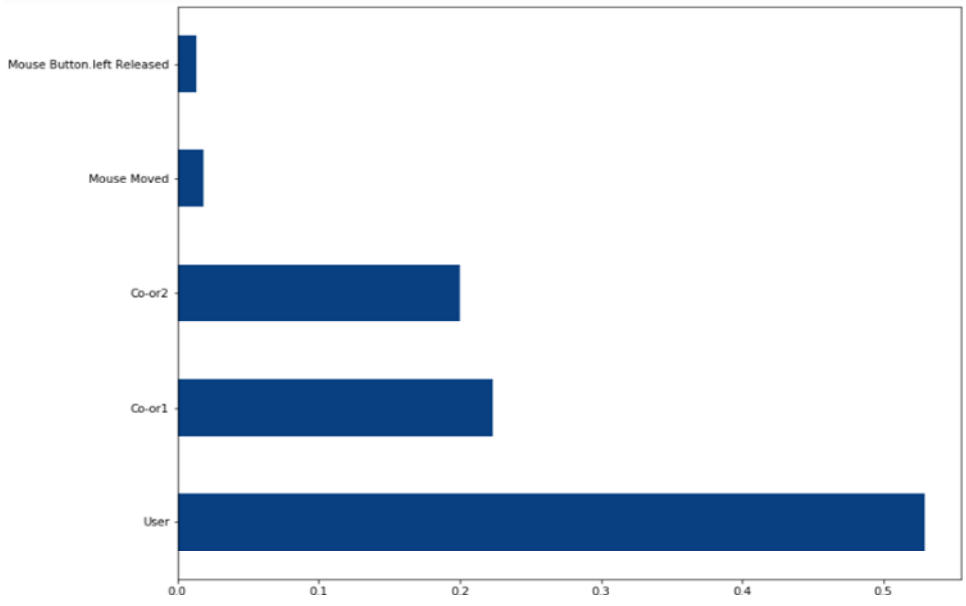
**Figure 4** Correlation matrix for mouse prediction data (see online version for colours)




The top 5 attributes needed for prediction were found using the extra tree classifier and the results are shown in Figure 5. The most important attribute was found to be the user attribute followed by the coordinates between which the mouse was moved.

- *Training:* The model was trained with many classifiers to find out which provided the best results without overfitting. The dataset was first split into training and testing data using sklearn's train_test_split method. The classifiers that were experimented with are SVM classifier, decision tree classifier, and the extra tree classifier. The three classifiers were first imported from the sklearn library and then applied to the dataset. Through the experimentation, it was found that the best results were obtained by the extra tree classifier.

**Figure 5** Top 5 attributes for mouse prediction data (see online version for colours)



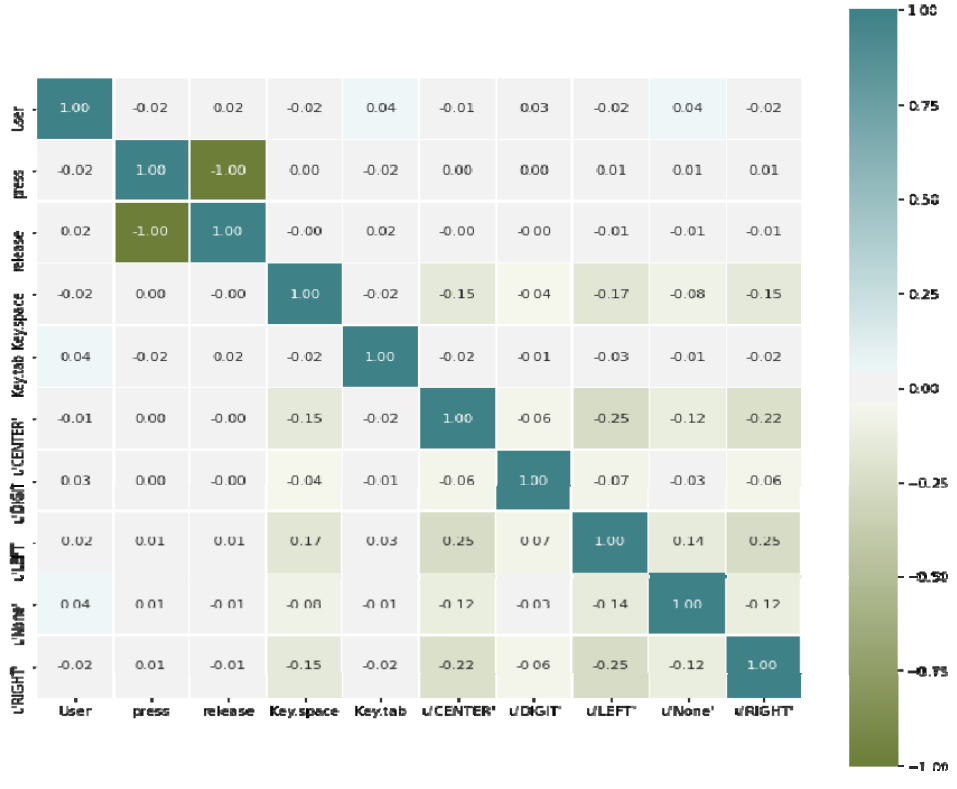

## *4.4 Keyboard prediction module*

This prediction module uses keystroke logging of the users to find out if any malicious user has logged in to the company's system, it is observed that a malicious user has a certain pattern which could be identified by a machine learning model so that when a malicious activity is carried out the authorised people could be notified about it on time. For the implementation of the keyboard prediction model, the following dataset and model was used:

- *Dataset:* The dataset was collected from an experiment and consisted of around 5 lakh timestamped keystroke logging of eight users, the attributes of the dataset were the timestamp of the activity, direction of the key (released or pressed), key pressed (left, right or centre) and the current user and the target value which tells if the user is a traitor masquerade or normal.
- *Data pre-processing and EDA:* Since the data was collected from an experiment, the timestamp was biased and was not helping in finding out malicious activity, so therefore timestamp was dropped, all numerical values were converted to categorical values with the help of label encoder and one hot encoding. Correlation matrix gave an insight on how the parameters were related to each other. It was plotted using a heat map to get better insights on it shown in Figure 7. The correlation matrix of the dataset was found, and the correlation plot is as shown in Figure 6.
- *Training:* Multiple models were trained on the dataset. 70% of the data was used for training and the remaining was used for testing, after performing hyper parameter tuning on various models, best results were given by SVM and CatBoostClassifier. SVM with its default parameters gave an accuracy of 54.64% and CatBoostClassifier

with learning rate 0.50 when run for 50 iterations gave an accuracy of 55.54%. Because of the slight increase in the accuracy, CatBoostClassifier was finally chosen. Since the dataset was collected from an experiment the accuracy of the model seems less, if real time stamped data of malicious users are obtained then an increase in the accuracy is expected. A Kivy-based front-end was also built to invoke the monitoring system and start the prediction module following which if an attack was detected the appropriate results (input parameter and outputs) are displayed to the user.

**Figure 6** Correlation matrix for keyboard data (see online version for colours)




## 5 Results

The intruder or malicious user detection systems used in the industry are traditional and need manpower to monitor and respond. In terms of the algorithms which have been trained via machine learning, the training and testing time due to the large size of the datasets is a major cause of inhibition in practical fields. Also, creation of large datasets that require manual labelling are intensive tasks difficult to achieve. Eliminating these constraints, a machine learning which provides high accuracy is chosen to train. The

Kivy-based frontend allowed us to ensure easy incorporation of the machine learning model that is trained and bringing it with an easy to use and understand interface to the user end.

The mouse prediction module refers to the mouse movements made by the user. All the mouse movements during usage are recorded and saved to a text file. Through the data collected in the text file as shown in Figure 7.

**Figure 7** Data collected for mouse trace (see online version for colours)

```
In [21]: df.tail(10)
```

Out[21]:

| | Timestamp | Direction | Co-or1 | Co-or2 | User | time | target |
|---|---|---|---|---|---|---|---|
| 1434086 | 2017-03-24 12:40:11,727 | Mouse Moved | 844.0 | 465.0 | User10 | NaN | N |
| 1434087 | 2017-03-23 12:05:08,846 | Mouse Moved | 101.0 | 332.0 | User11 | NaN | M |
| 1434088 | 2017-03-24 15:48:52,743 | Mouse Moved | 1130.0 | 257.0 | User10 | NaN | N |
| 1434089 | 2017-03-24 20:14:55,983 | Mouse Moved | 300.0 | 121.0 | User12 | NaN | N |
| 1434090 | 2017-03-22 01:19:38,828 | Mouse Moved | 803.0 | 260.0 | User1 | NaN | T |
| 1434091 | 2017-03-24 10:18:58,259 | Mouse Moved | 455.0 | 269.0 | User12 | NaN | N |

The keyboard prediction module uses keystroke logging of the users to find out if any malicious user has logged in to the company's system, it is observed that a malicious user has a certain pattern which could be identified by a machine learning model so that when a malicious activity is carried out the authorised people could be notified about it on time and the data collected is shown in Figure 8.

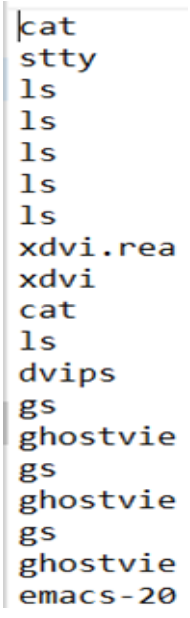
**Figure 8** Data collected for keyboard trace (see online version for colours)

```
df.head()
```

| | Timestamp | User | target | press | release | Key.space | Key.tab | u'CENTER' | u'DIGIT' | u'LEFT' | u'None' | u'RIGHT' |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2017-03-23 01:58:41,631 | 5 | M | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 2017-03-21 20:29:48,529 | 1 | M | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 2 | 2017-03-23 20:27:04,407 | 1 | M | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 3 | 2017-03-24 10:53:32,115 | 1 | N | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 4 | 2017-03-20 16:40:22,549 | 5 | N | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

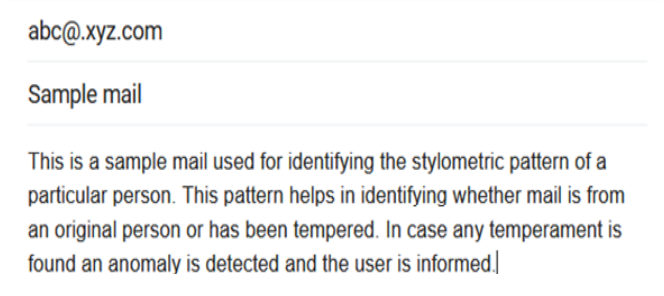
Command trace is essentially monitoring all the commands issued by the user to the system. This module traces all the commands and saves it into a text file as shown in Figure 9.

**Figure 9** Data collected for command trace




Authorship identification pertains to discovering the author of unknown documents as shown in Figure 10. Whenever there is an incident of insider attack, when considering organisational security or cases like cyber bullying authorship attribution can play a major role in identifying the perpetrator and said digital forensics.

**Figure 10** Data collected for authorship attribution




The dataset being used consists of a log of the different operations carried out using a mouse. The total number of logs in the dataset is over 1.4 million. Collected mouse data is visualised in 2D as shown in Figure 11.

Different mouse movements made by the user are considered. The attributes of the dataset are timestamp of the mouse activity, type of activity (mouse pressed, mouse moved, etc.), coordinates of the mouse, the user logged in, and finally the target attribute which tells if the mouse activity is normal, masquerade or a traitor. Collected mouse data is visualised in 3D as shown in Figure 12.

**Figure 11** Visualising mouse data in two-dimension (see online version for colours)

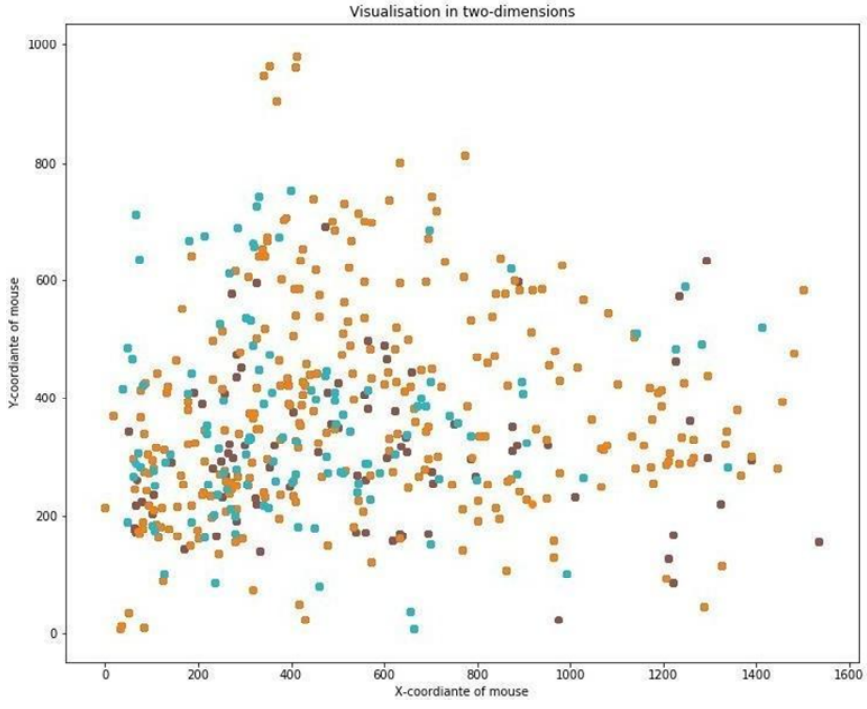
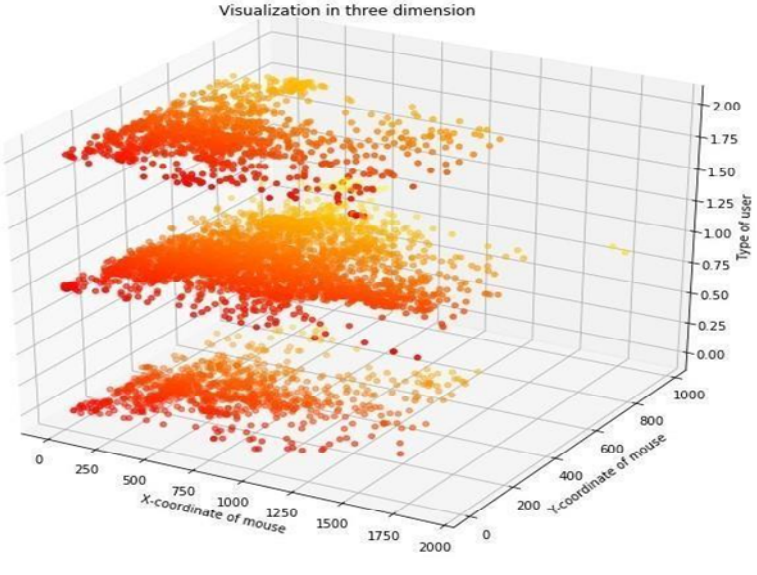



**Figure 12** Visualising mouse data in three-dimension (see online version for colours)




Since the data was collected from an experiment the timestamp was biased and was not helping in finding out malicious activity, so therefore timestamp was dropped, all

numerical values were converted to categorical values with the help of label encoder and one hot encoding. Label encoding was performed on the attributes user and target, and one hot encoding was done on the attribute direction. All these features are plotted as shown in Figure 13.

**Figure 13** Plotting most important features (see online version for colours)

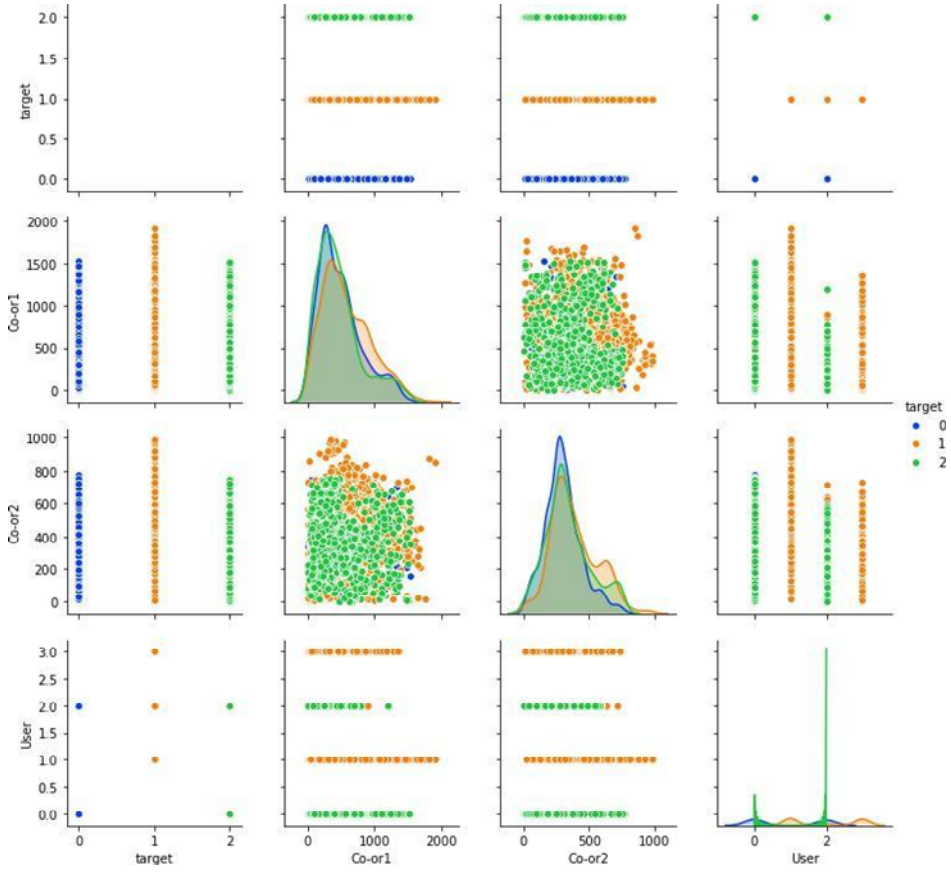


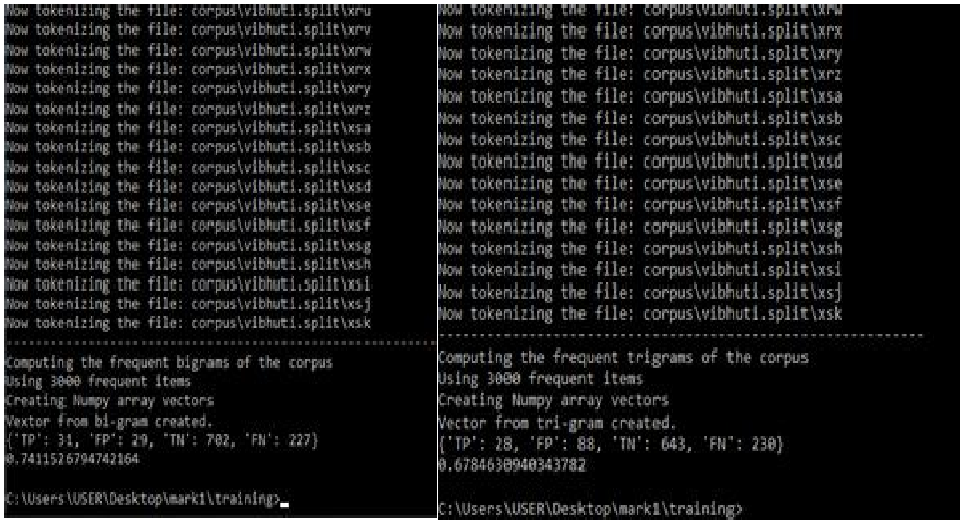

**Table 1** Accuracy score for different authors

| *S. no.* | *Authors* | *Accuracy from different machine learning algorithms* | | | |
|---|---|---|---|---|---|
| | | *Support vector machine (%)* | | *Naive Bayes (%)* | |
| | | *Bi-gram* | *Tri-gram* | *Bi-gram* | *Tri-gram* |
| 1 | Author 1 | 94.48 | 94.37 | 87.96 | 88.30 |
| 2 | Author 2 | 64.91 | 73.91 | 74.11 | 67.84 |
| 3 | Author 3 | 75.28 | 74.39 | 83.01 | 82.25 |
| 4 | Author 4 | 79.86 | 79.86 | 65.24 | 76.82 |

The accuracy score for four different authors have been calculated with the help of SVM and NB machine learning algorithms as shown in Table 1.

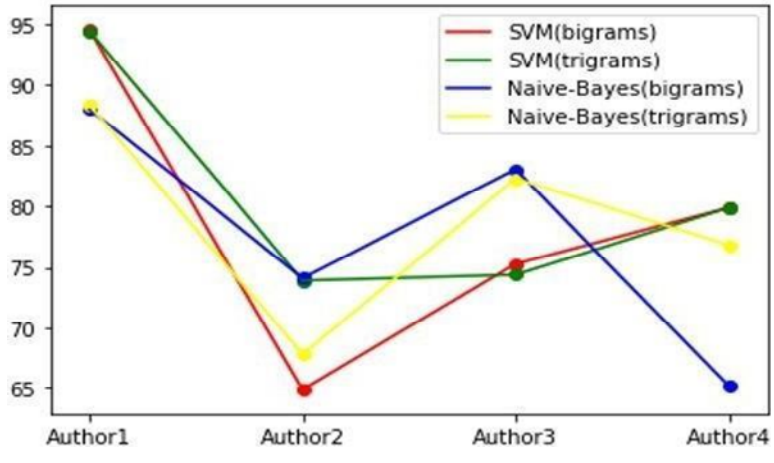
The bi-grams and tri-grams of Author 3 are computed as shown in Figure 14 using frequent items.

**Figure 14** Results for Author 3 using bi-grams and tri-grams




The accuracy is plotted in the y-axis and four different authors in the x-axis for two different machine learning algorithms to know the minimum and maximum accuracy as shown in Figure 15. SVM and NB algorithms were run on the four different authors. The line graph shows that the minimum accuracy was attained by the Author 2. The accuracy score is as indicated in Figure 14. SVM (bi-grams) shows minimum accuracy.

**Figure 15** Plotting minimum and maximum accuracy (see online version for colours)




The mouse traces, collected from the mouse movement, which will be passed through a decision tree classifier for knowing the accuracy of both train and test sets as shown in Figure 16.

**Figure 16** Accuracy score for mouse trace (see online version for colours)

```
y_train_pred = clf.predict(X_train)
print(accuracy_score(y_train_pred, y_train))
```

```
0.9824571428571428
```

```
from sklearn import tree
clf_tree = tree.DecisionTreeClassifier()
clf_tree.fit(X_train, y_train)
y_test_tree = clf_tree.predict(X_test)
print("Accuracy of test set:",accuracy_score(y_test_tree, y_test))


y_train_tree = clf_tree.predict(X_train)
print("Accuracy of train set:",accuracy_score(y_train_tree, y_train))
```

```
Accuracy of test set: 0.7706666666666667
Accuracy of train set: 0.9997142857142857
```

The keystrokes collected from the keyboard data, which will be passed through an SVM classifier to calculate the accuracy as shown in Figure 17.

**Figure 17** Accuracy score for keyboard trace (see online version for colours)

```
clf = SVC()
clf.fit(X_train, y_train)
```

```
C:\Users\singh\Anaconda3\lib\site-packages\sklearn\svm\base.py:193: Futu
'auto' to 'scale' in version 0.22 to account better for unscaled feature
his warning.
  "avoid this warning.", FutureWarning)
```

```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto_deprecated',
    kernel='rbf', max_iter=-1, probability=False, random_state=None,
    shrinking=True, tol=0.001, verbose=False)
```

```
y_pred = clf.predict(X_test)
print(sum(y_pred==y_test))
```

```
8196
```

```
print(accuracy_score(y_pred,y_test))
```

```
0.5464
```

Command trace is essentially monitoring all the commands issued by the user to the system. This module traces all the commands and saves it into a text file. These monitored commands will then be used for training the machine learning model which will predict any deviation from generic behaviour pattern or any malicious behaviour taking place.

For training a machine learning model, one has KNN since the goal is to classify data into two categories of malicious or non-malicious user. Therefore, a KNN classifier from sklearn library is imported. The model is divided into testing and training subsets respectively and trains the model on training dataset and accuracy is improved after hyper-parameter tuning as shown in Figure 18.

**Figure 18** Accuracy score for command trace after hyper-parameter tuning

```
0.98
0.9533333333333334
0.9866666666666667
1.0
1.0
0.9333333333333333
1.0
0.84
0.96
```

Figure 19 shows the final application screenshot of the online tool to test authorship of any document. Here, selected Author 9: XYZ.

**Figure 19** Online tool to test authorship of any document (see online version for colours)

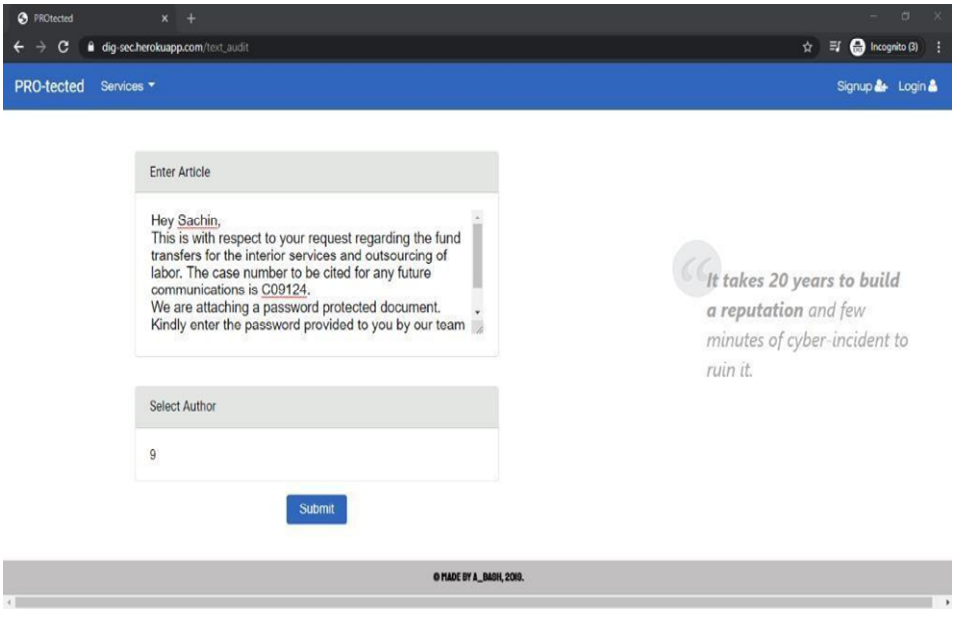



Figure 20 shows authentication failure of Author 19 with probability 81%.

**Figure 20** Result window (here: authentication fails) (see online version for colours)

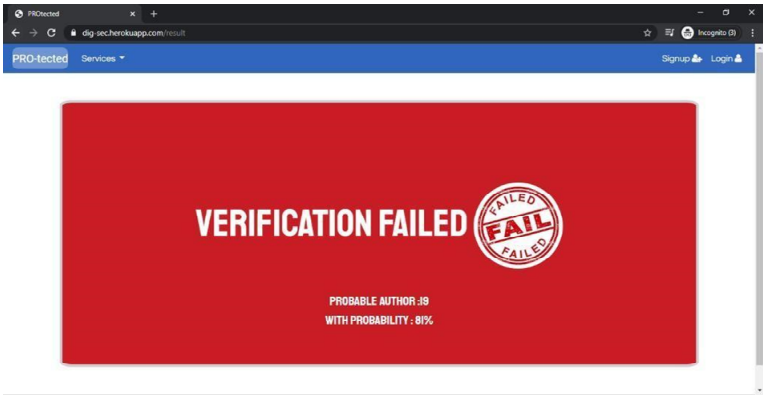



**Figure 21** Result window (here: clearing authentication) (see online version for colours)

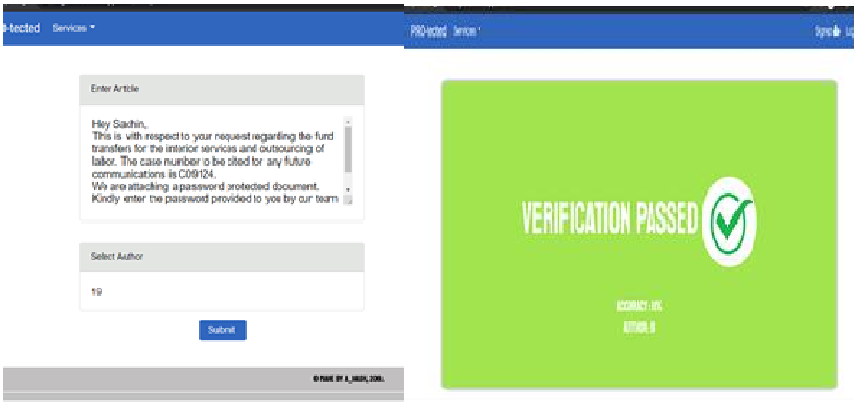



**Figure 22** Desktop application opening screen (see online version for colours)

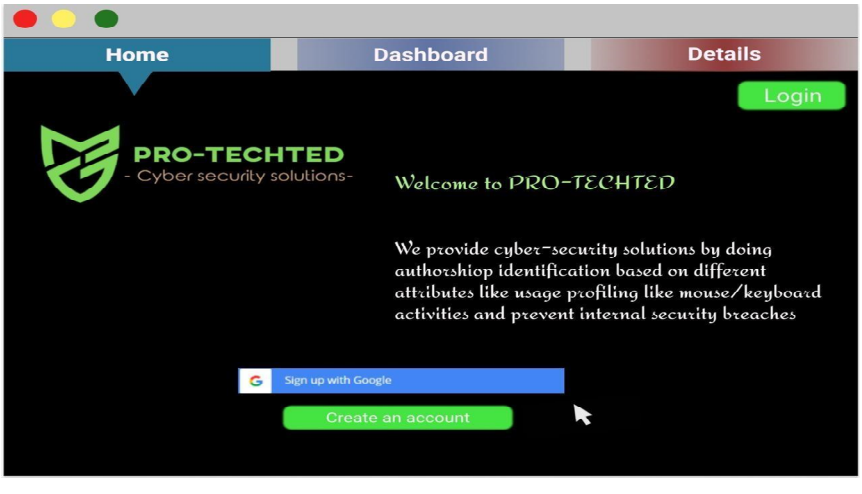


Figure 21 shows clearing authentication of Author 19 with probability of 81%.

Figure 22 shows the desktop opening screen. One can sign in using a Google account.

Figure 23 shows the dashboard of the web application where active users and user details and the danger levels can be known.

**Figure 23** Dashboard for monitoring activity (see online version for colours)

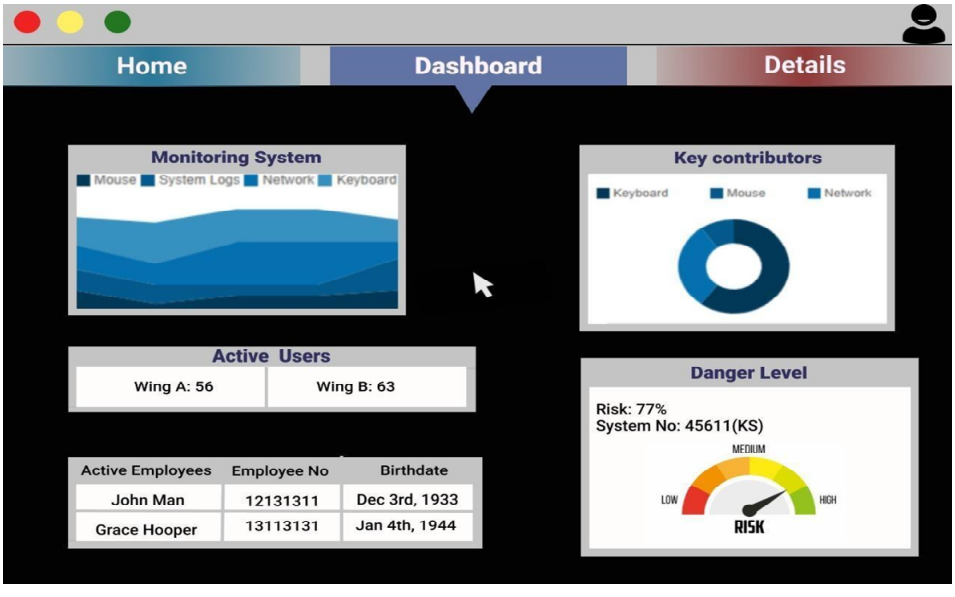



**Figure 24** Window for analysing details of the anomaly (see online version for colours)

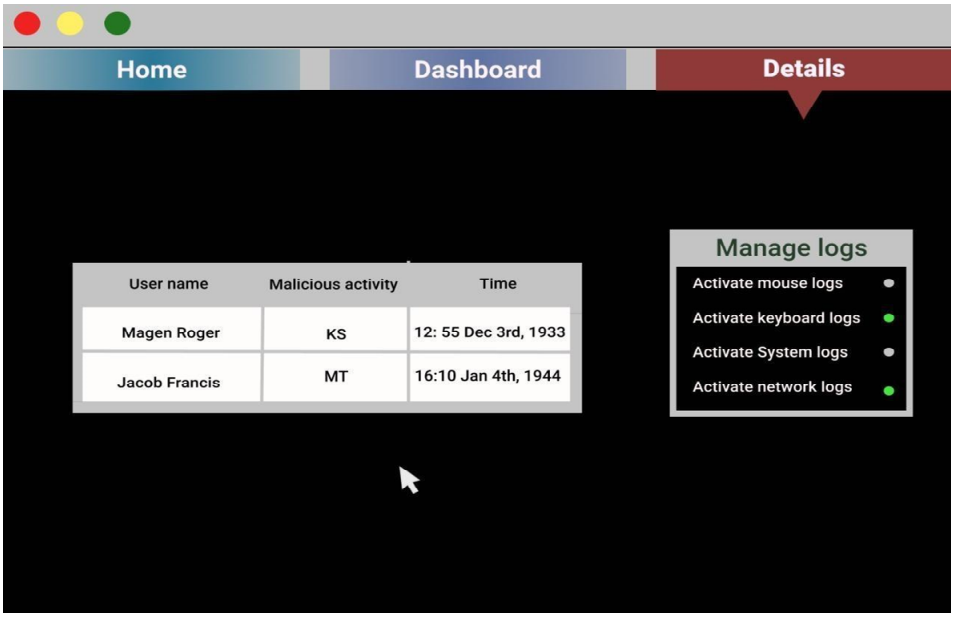


Figure 24 shows the window for analysing the details of the anomaly with the user details and the managed log details.

## 6 Conclusions

The individual modules are trained on authorship attribution, mouse monitoring, keyboard monitoring and command tracing and reached promising results with good accuracies in the range of 65%–85% on an average. Hence, it is clear that the humongous threat of insider attack and cyber security in a company can be resolved with the aid of state-of-the-art machine learning techniques. One can train SVM and NB classifiers for Hindi language which also shows that machine learning and language processing are equally suitable for regional languages and these solutions can be deployed. As a future work, one can work on enhancing the accuracy of these modules such that the results generated become more thorough and definitive and any threat can be handled on time. Further cyber bullying which takes advantage of online anonymity can be resolved if the investigators narrow down a set of suspects and their general posting style is passed through the authorship attribution module.

## References

Alhijawi, B., Hriez, S. and Awajan, A. (2018) 'Text-based authorship identification – a survey', in *2018 Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT)*, IEEE, pp.1–7.

Basodi, S., Tan, S., Song, W. and Pan, Y. (2020) 'Data integrity attack detection in smart grid: a deep learning approach', *International Journal of Security and Networks*, Vol. 15, No. 1, pp.15–24.

Bozkurt, I.N., Baglioglu, O. and Uyar, E. (2007) 'Authorship attribution', in *2007 22nd International Symposium on Computer and Information Sciences*, IEEE, pp.1–5.

Chen, G., Shi, X., Chen, M. and Zhou, L. (2020) 'Text similarity semantic calculation based on deep reinforcement learning', *International Journal of Security and Networks*, Vol. 15, No. 1, pp.59–66.

Dauber, E., Overdorf, R. and Greenstadt, R. (2017) 'Stylometric authorship attribution of collaborative documents', in *International Conference on Cyber Security Cryptography and Machine Learning*, Springer, Cham, pp.115–135.

Dugar, T.K., Gowtham, S. and Chakraborty, U.K. (2019) 'Hyperparameter tuning for enhanced authorship identification using deep neural networks', in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, IEEE, pp.206–211.

Elmasry, W., Akbulut, A. and Zaim, A.H. (2018) 'Deep learning approaches for predictive masquerade detection', *Security and Communication Networks 2018*.

Harilal, A., Toffalini, F., Homoliak, I., Castellanos, J.H., Guarnizo, J., Mondal, S. and Ochoa, M. (2018) 'The wolf of SUTD (TWOS): a dataset of malicious insider threat behavior based on a gamified competition', *JoWUA*, Vol. 9, No. 1, pp.54–85.

Hurtado, J., Taweewitchakreeya, N. and Zhu, X. (2014) 'Who wrote this paper? Learning for authorship de-identification using stylometric featuress', in *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)*, IEEE, pp.859–862.

Kallimani, J.S., Chandrika, C.P., Singh, A. and Khan, Z. (2019) 'Authorship identification using supervised learning and n-grams for Hindi language', *ICRITCSA Conference*.

Nirkhi, S.M., Dharaskar, R.V. and Thakare, V.M. (2015) 'Authorship identification using generalized features and analysis of computational methods', *Transactions on Machine Learning and Artificial Intelligence*, Vol. 3, No. 2, p.41.

Oladimeji, T.O., Ayo, C.K. and Adewumi, S.E. (2019) 'Review on insider threat detection techniques', *Journal of Physics: Conference Series*, Vol. 1299, No. 1, p.012046, IOP Publishing.

Raghavan, P. and El Gayar, N. (2019) 'Fraud detection using machine learning and deep learning', in *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, IEEE, December, pp.334–339.

Salem, M.B., Hershkop, S. and Stolfo, S.J. (2008) 'A survey of insider attack detection research', in *Insider Attack and Cyber Security*, pp.69–90, Springer, Boston, MA.

Sánchez-Aguayo, M., Urquiza-Aguiar, L. and Estrada-Jiménez, J. (2021) 'Fraud detection using the fraud triangle theory and data mining techniques: a literature review', *Computers*, Vol. 10, No. 10, p.121.

Shaukat, K., Luo, S., Chen, S. and Liu, D. (2020) 'Cyber threat detection using machine learning techniques: a performance evaluation perspective', in *2020 International Conference on Cyber Warfare and Security (ICCWS)*, IEEE, October, pp.1–6.

Swain, S., Mishra, G. and Sindhu, C. (2017) 'Recent approaches on authorship attribution techniques – an overview', in *2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA)*, IEEE, Vol. 1, pp.557–566.

Yan, Y., Zhang, L.X., Wang, B.Q. and Gao, X. (2020) 'Location big data differential privacy dynamic partition release method', *International Journal of Security and Networks*, Vol. 15, No. 1, pp.25–35.