EMITTER International Journal of Engineering Technology
Vol. 11, No. 2, December 2023, pp. 125~144
DOI : 10.24003/emitter.v11i2.804

KFREAIN: Design of A Kernel-Level Forensic Layer for Improving Real-Time Evidence Analysis Performance in IoT Networks

Seema Shukla[1], Sangeeta Mangesh[1], Prachi Chhabra[2]

[1]Dronacharya Group of Institutions, Greater Noida, Uttar Pradesh, India
[2]JSS Academy of Technical Education, Noida, Uttar Pradesh, India
Corresponding Author: seema.shukla@gnindia.dronacharya.info

Received September 9, 2023; Revised October 15, 2023; Accepted December 8, 2023

Abstract

An exponential increase in number of attacks in IoT Networks makes it essential to formulate attack-level mitigation strategies. This paper proposes design of a scalable Kernel-level Forensic layer that assists in improving real-time evidence analysis performance to assist in efficient pattern analysis of the collected data samples. It has an inbuilt Temporal Blockchain Cache (TBC), which is refreshed after analysis of every set of evidences. The model uses a multidomain feature extraction engine that combines lightweight Fourier, Wavelet, Convolutional, Gabor, and Cosine feature sets that are selected by a stochastic Bacterial Foraging Optimizer (BFO) for identification of high variance features. The selected features are processed by an ensemble learning (EL) classifier that use low complexity classifiers reducing the energy consumption during analysis by 8.3% when compared with application-level forensic models. The model also showcased 3.5% higher accuracy, 4.9% higher precision, and 4.3% higher recall of attack-event identification when compared with standard forensic techniques. Due to kernel-level integration, the model is also able to reduce the delay needed for forensic analysis on different network types by 9.5%, thus making it useful for real-time & heterogenous network scenarios.

Keywords: IoT, Kernel Layer, TBC, BFO, Forensics.

1. INTRODUCTION

There are several examples of networked devices [1, 2]: smart vehicles like unmanned drones and driverless cars, smart appliances, smart household companion systems like Amazon Echo and Google Home, and Web of Battlefield / Military Things devices, to name a few. Examples of smart vehicles include autonomous autos and aerial devices that use Fuzzy Hashed Blockchains (FHB) [1, 2, 3]. [2] Smart refrigerators include home assistant technologies like Amazon Echo and Google Home. These devices, which also include Internet of Things (IoT) devices, generate a lot of data, and that data may be quickly transported from one or more source devices to other

connected devices or systems [4, 5, 6]. The data or systems holding it may then be targeted in attacks, often by people with ulterior motivations like financial gain (e.g., selling of data exfiltrated from compromised systems). It is essential to have the tools necessary to conduct a comprehensive analysis of the impacted digital systems and devices. Digital forensics is the process of completing an in-depth analysis of digital devices and data in the context of a legal proceeding, such as a criminal or civil investigation via use of Hardware based Neural Frameworks (HNF) [7, 8, 9]. Our capacity to analyze all the various storage capacities and devices fast is being hampered by the difficulty of doing so. The enormous amount of information produced by computers and other electronic devices, known as digital forensic data, have been the subject of fierce debate for years [5]. Each year, there is an increase in the volume of digital forensic data as well as the types of data that are available for forensic analysis [6, 7]. This may be due to the proliferation of devices, their variety, and the data they generate. Digital forensic data may be compressed using our recently released data reduction technique without sacrificing any of the data included in the metadata or the original source file format [10, 11, 12]. The recent release of it for general use was proof of this. Numerous digital forensic and analysis tools, including EnCase, X-Ways Forensic, NUIX, Magnet Forensic Internet Evidence Finder (IEF), Intella, and Access Data FTK, may be used to analyze and investigate the data subsets [13, 14, 15, 16]. Access Data FTK and NUIX are two other well-known business products. The diverse subsets of the data are stored in common forensic logical containers (L01) [17, 18, 19, 20]. Digital forensic samples may also be examined using a range of programs and analytical instruments by mounting them as logical drives. For instance, important apps in this area include RegRipper, Windows File Analyzer, and NetAnalysis [21, 22, 23, 24]. This puts us in a situation where data may be aggregated and mixed for analytical purposes from a wide variety of devices, including data from mobile phones and data saved in the cloud via Federated Learning and Network Traffic Feature Engineering (FL NTF) [25, 26, 27, 28]. This opens up a ton of fresh possibilities. Combining data from several sources is a strategy that may help in digital forensic analysis [29, 30, 31, 32]. This is due to the possibility that connections between events that first seemed unconnected and various pieces of equipment might sometimes provide light on a mysterious data set samples [33, 34, 35, 36]. This may lead to discoveries or the development of new lines of inquiry, both of which might hasten the process of concluding criminal investigations or revive closed case files with new evidence sets [37, 38, 39, 40]. This paper suggests the creation of a scalable Kernel-level Forensic layer, aiming to enhance the real-time analysis of evidence performance and facilitate efficient pattern analysis of collected data samples.

2. RELATED WORKS

Due to the rapid evolution of kernel types, the development of designs that are general purpose is an essential component of kernel design. In order

to accomplish this goal, a wide variety of various sorts of system models have been developed. One illustration of this would be how the authors of [4] propose utilizing a support vector machine (SVM) model to choose kernels from a group of Internet of Things machine functions. This enables the system to choose kernels that are cognizant of latency, energy consumption, and performance in accordance with the requirements of the application. The research suggests the development of a tool for auditing at the kernel level that can evaluate the performance of the network at the kernel level in real time. It is possible for the kernel of the IoT network to implement these quality of service measures if it can determine the location of the nodes, their energy consumption, and a number of other characteristics. For instance, the Android kernel was modified in [6] in order to improve the way it manages energy. For the purpose of assisting with restricted Internet of Things devices, a microkernel architecture is developed, as shown in [7], and this design may be tweaked for increased performance. The adaptability of this design makes it possible to realize gains in terms of the overall efficiency of the network. The deep models that are presented in the work in [8] have exceptional performance efficiency, but their initial energy consumption is substantial, which makes them unsuitable for use in IoT kernels. It has been shown in [9] and [10] that these methods might reduce their processing requirements by shifting responsibilities to other individuals. These methods could make use of the built-in kernel offloading resources available to them. Clustering of Internet of Things nodes and egocentric graph mining are the two methods that these models use to accomplish this goal. In [11], high-performance IoT kernels are used in order to classify photo sets by the application of these models.

As suggested in [12], enhancing the security of the IoT kernel might be accomplished by merging various fuzzy logic methodologies with Binary Static Analysis. The kernel may, with the assistance of this evaluation, assign a one-of-a-kind zone for each embedded IoT microdevice, therefore segregating the activity of these microdevices from that of other processes. The inclusion of sandbox behavior in this way immediately results in a boost in network performance. According to [13], the ability to forecast events could lead to even greater improvements in safety and security. The SVM and latent Dirichlet allocation (LDA) algorithms allow for the prediction of a wide variety of occurrences, such as the failure of a node or connection, among other things. For the purpose of improving throughput, super-resolution processing, phase detection, and security applications, it is advised that models similar to those described in [14], [15], [16], and [17] be used. Previous research [18], [19], [20], and [21] suggests that the present kernels of the Internet of Things might be updated to give security verification, threading for performance and security, domain parameter prediction for improved threat mitigation, and high speed IoT performance. Altering the kernels used in this situation, such as TizenOS, Real time kernel, etc., results in the introduction of improvements.

Models such as Sparse Approximation [22], GPU-based Collaborative Filtering [23], Metric-Chisini-Jensen-Shannon Divergence [24], Multi-Kernel and Meta-heuristic Feature Selection [25], and Convolutional-Kernel-Readout Method [26] have the potential to further improve the performance of the Internet of Things. These methods make use of the multithreading and parallel processing capabilities of the kernel, which allows them to accomplish the task in a shorter amount of time. Hyperspectral remote sensing [27], uplink narrowband data rate improvement [28], ultra-low power system-on-chip (SoC) architecture [29], and ubiquitous electric power [30] are just a few examples of applications that might potentially benefit from the use of these powerful techniques. The kernel has been modified in a way that makes this possible by shifting execution away from the CPU and onto the underlying hardware. The overall performance of the system is improved when the processing levels are lowered. According to [31], putting the kernel's security at risk may be possible if a malicious kernel implementation had the capability to decode assembly-level applications and other binaries. This is because such an implementation might potentially compromise the integrity of the kernel. This might lead to severe concerns when applying IoT kernels in sensitive applications such as the one for disaster prediction outlined in [32]. These kinds of defects might lead to incorrect data reporting, which would make it more difficult for the operator to take corrective action. The implementation of learning models that are suitable for the Internet of Things, as outlined in [33], is one possibility among many others. These models may be pre-programmed to recognize potentially harmful behaviors and provide an alarm to the system administrator, who would then be able to take the appropriate actions. Additional Internet of Things kernel applications are shown in references [34], [35], [36], and [37]. These applications include intrusion detection, computation using a Coordinate Rotation Digital Computer (CORDIC), separation kernel security, and extension of access. These applications' kernel settings have been modified in order to facilitate rapid data processing. The reconfigurability of Internet of Things devices is improved with the help of ChamelIoT kernel's [38] sophisticated monitoring and control capabilities. The performance of sentiment analysis is improved by using enhancements to the 6G network as well as high-throughput operations, as shown in [39] and [40]. [41] provides information on Zephyr OS, which is one of the most well-known kernels for IoT devices. This operating system is utilized in the underlying research for the purpose of modification and comparison. This is due to the fact that it has a high degree of flexibility and is suitable for adoption by a broad variety of IoT manufacturers. Sandboxing and buffer management are two methods that are suggested in [42, 43] as potential strategies to improve these kernels. Work in [44, 45, 46, 47] also provides a set of deep learning frameworks for improving security performance under different network conditions. Work in [48, 49, 50] provides incremental learning models for enhancing classification accuracy levels. While work in [51, 52, 53, 54] provide deep learning models to perform

incremental learning operations for different network scenarios. It is evident that very little research has been done on the topic of modifying the kernel to enhance the quality of service and security levels.

While hardware-based neural frameworks have the potential to accelerate digital forensics investigations and improve efficiency, their practical implementation requires careful consideration of hardware compatibility, data privacy, cost, and scalability. As digital forensic tools and technologies continue to advance, hardware acceleration may become a crucial component in the investigator's toolkit for handling the ever-increasing volume and complexity of digital evidence. Combining Federated Learning and Network Traffic Feature Engineering in Digital Forensics provides Enhanced privacy, collaborative intelligence, efficient investigations, and automated evidence identification. However, the approach has disadvantages such as communication overhead, data heterogeneity, security risks, and coordination complexity may hinder efficiency and transparency. Some of the pros of FHB are robust data integrity and tamper-resistant evidence with transparent verification and reliable data provenance while the cons are computational overhead and adoption challenges, scalability concerns, and evolving regulatory frameworks may pose implementation hurdles.

3. ORIGINALITY

It is clear, that academics have discussed several different machine learning-based model types [41, 42, 43, 44]. When tested against real attacks, most of these models perform at the application level, which reduces their effectiveness. It is almost hard to adapt most of these models to account for a broader variety of variables since they were designed for certain kinds of networks [45, 46]. The creation of a Kernel-level Forensic layer will be discussed as a potential remedy to these problems in the next section. This layer will aid in enhancing real-time evidence processing performance across a range of diverse network circumstances. In this research, we examine the use of various hardware and data subsets with reduced processing and storage needs. Utilizing cloud-sourced data, device data, data reduction, and quick analytic approaches, our objective is to assess the value of cross-device and cross-case analysis. The effectiveness of the suggested model is assessed and contrasted with the normative practices used in forensic investigations in the section 5 of this text.

4. SYSTEM DESIGN

Figure 1 illustrates the design of a scalable kernel-component that is compatible with the majority of modern IoT kernels and aids in the efficient pattern analysis of collected data samples. The proposed kernel component includes a Temporal Blockchain Cache (TBC) that is refreshed after analyzing each evidence set. The model proposes using a multidomain feature extraction engine that combines lightweight Fourier, Wavelet, Convolutional, Gabor, and Cosine feature sets to perform this analysis. A stochastic Bacterial Foraging

Optimizer (BFO) selects these feature sets, which aids in the identification of high variance features. The selected features are then processed by an ensemble learning (EL) classifier comprised of the Nave Bayes (NB), k Nearest Neighbor (kNN), Logical Regression (LR), and Multilayer Perceptron (MLP) classifier sets.

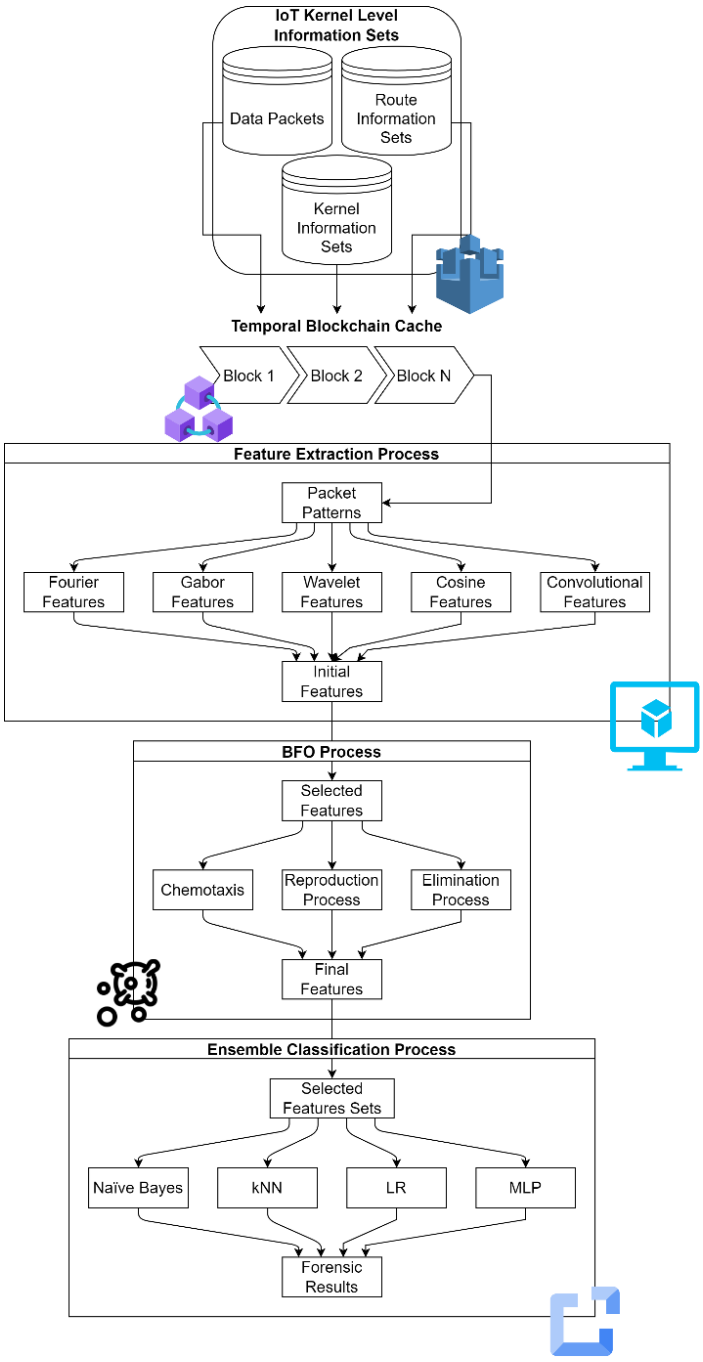

Figure 1. Design of the Proposed Kernel-level Forensic Layer

4.1 Temporal Blockchain Cache (TBC)

To enable kernel-level immutability of packets, traceability of data samples, high speed during retrieval operations and distributed computing capabilities, the model uses a Proof-of-Trust (PoT) based blockchain for storing packets. The block structure used to store the PoT packets can be observed from Table 1 where different kernel-level information sets are stored for future analysis.

Table 1. Block structure used for storing network packets

Prev. Hash	Orig. IP	Dest. IP
Timestamp	Sample Sets	Meta Data Sets
Kernel-level information sets	Nonce & Trust Information Sets	Current Hash

The information stored in this block structure includes hash of the previous blocks (P_{hash}), IP of originating node and destination nodes (IP), Timestamp of the packets (TS), a stochastic nonce value, Meta Data ($Meta$) and Kernel-level information about the packets and current hash of the block. The stochastic nonce value, is generated via Eq. 1,

$$nonce = STOCH\big(1, Max(N)\big) \tag{1}$$

where, N represents data-type ranges, and $STOCH$ represents a Markovian stochastic process used to generate number sets. The Meta Data ($Meta$) and Kernel-level information about the packets includes information such as packet identifiers, flow rate of the packets, range of values in the packets and hashing & encryption constants. The current hash of the block is calculated using Secure Hashing Algorithm (SHA256) via Eq. 2,

$$B_{hash} = SHA256(P_{hash}|IP|SS|T_s|Meta|nonce) \tag{2}$$

where, $P_{hash}, B_{type}, B_{data}, and\ T_s$ represents previous hash, biometric type, biometric data, and timestamp of block creation. The hashes are generated by reiterating Eq.1 & 2, so that every block hash has unique hash sets. After generation of unique hashes, blocks are added to the chain through PoT based consensus. The PoT based consensus model evaluates trust-levels for neighbouring nodes via Eq. 3,

$$TL_i = \frac{1}{NM}\sum_{j=1}^{NM}\frac{d_j}{Max(d)} + \frac{e_j}{Max(e)} + \frac{Max(THR)}{THR_j} + \frac{100}{PDR_j} \tag{3}$$

where, d, e, THR & PDR represents the delay needed, energy consumed, throughput achieved, and packet delivery ratio achieved during previous NM mining requests. This trust level is estimated for all 1-hop neighbouring nodes, and then average TL is estimated via Eq. 4,

$$TL_{th} = \frac{1}{N(1Hop)} \sum_{i=1}^{N(1Hop)} TL_i \tag{4}$$

where, $N(1Hop)$ represents total 1Hop nodes present near to the current node that has generated mining requests. Hashes from nodes with $TL < TL_{th}$ are considered, and new blocks are added to current blockchains. Once the blockchain is created, and forensic are requested for a particular node, then their data samples are collected via hash-matching process. These data samples are stored on a similar blockchain temporarily, which assists in creation of a set of Temporal Blockchain Caches (TBCs).

4.2 Feature Extraction Process

Samples stored on TBCs are converted into multidomain features ensuring that data confidentiality is maintained while processing, because it is not possible to regenerate original sample values from the aggregated feature sets and accuracy of classification is improved due to use of highly variant features for analysis. To perform this task, initially the collected samples are represented into frequency domain using Fourier analysis via Eq. 5,

$$DFT_i = \sum_{j=1}^{N_f} x_j * \left[cos\left(\frac{2 * \pi * i * j}{N_f}\right) - \sqrt{-1} * sin\left(\frac{2 * \pi * i * j}{N_f}\right)\right] \tag{5}$$

where, N_f represents total number of samples extracted for current set of forensic nodes. This feature assists in identification of repetitive analysis behaviour for different packet types. These features are extended via use of entropy-based cosine components, which are extracted via Eq. 6,

$$DCT_i = \frac{1}{\sqrt{2 * N_f}} * x_i \sum_{j=1}^{N_f} x_j * \cos\left[\frac{\sqrt{-1} * (2 * i + 1) * \pi}{2 * N_f}\right] \tag{6}$$

These components assist in identification of packet energy levels under different types of forensic events. Similarly, the convolutional features assist in identification of window-based overlapping features via Eq. 7,

$$Conv_{out_i} = \sum_{a=-\frac{m}{2}}^{\frac{m}{2}} x(i-a) * LReLU\left(\frac{m+2a}{2}\right) \quad (7)$$

where, m, a represents dimensions of different windows & strides, while $LReLU$ represents a leaky rectilinear unit that is used for activation of these features via Eq. 8,

$LReLU(x) = l_a * x, whe\ n\ x < 0$,

$$else\ LReLU(x) = x \quad (8)$$

where, l_a is a quantization scaling constant, which is used to remove negative feature sets. These features are further extended via evaluation of Gabor components via Eq. 9,

$$G(x,y)_s = e^{\frac{-x`^2+\partial^2*y'^2}{2*\emptyset^2}} * \cos\left(2 * \frac{pi}{\lambda} * x'\right) \quad (9)$$

where, x, y are the feature dimensions, while $\partial, \emptyset$ & λ represents angular and wavelength constants for augmentation of features. These features are extended via equations 10 & 11 for identification of approximate & detailed Wavelet components.

$$W_a = \frac{x_i + x_{i+1}}{2} \quad (10)$$

$$W_d = \frac{x_i - x_{i+1}}{2} \quad (11)$$

4.3 Bacterial Foraging Optimizer (BFO)

High variance feature sets are identified using Bacterial Foraging Optimization (BFO) technique. First all extracted features are combined to obtain a Forensic Feature Set (FFS) and then Bacterial Foraging Optimization (BFO) is performed on FFS. This selection assists the classifiers to efficiently identify different forensic events, and also introduces non-reversible characteristics into the extracted features. To implement the BFO Model the following Bacterium reconfiguration constants are used:

- Total Bacterium that will be initially generated and reconfigured (NB)
- Total reconfiguration steps or iterations that will be used to process these bacterium (NI)
- Bacterium social learning rates (L_r)
- Set of features which were extracted during multidomain representation operations (N_f)

To start the optimization process N features are stochastically selected from the list of multidomain features via Eq.12,

$$N = STOCH(L_r * N_f, N_f) \tag{12}$$

As per this selection, class-level variance is identified and marked as Bacteria fitness via Eq.13,

$$f = \sqrt{\frac{\sum_{a=1}^{m}(f_a - \frac{\sum_{i=1}^{m}\sqrt{\frac{\sum_{j=1}^{n}(f_j - \frac{\sum_{a=1}^{n} f_a}{n})^2}{n-1}}}{m})^2}{m-1}} \tag{13}$$

where, m, n are the features of current forensic-event class, and other forensic-event classes, such that $N = m + n$, while f_a is the value of the multidomain features. Based on these solutions bacteria chemotaxis threshold is calculated via Eq.14,

$$f_{th} = \sum_{i=1}^{N_s} f_i * \frac{L_r}{N_s} \tag{14}$$

Bacterium with $f > f_{th}$ are reproduced, and passed to the next iterations, while others are eliminated in the current set of iterations. Eliminated bacteria are regenerated as follows:

- Stochastic features are selected via Eq.12 & 13, where only $\frac{N}{2}$ new feature sets are used, while other $\frac{N}{2}$ are selected stochastically from reproduced bacterium solutions.
- For these N updated features, fitness is estimated via Eq.14, and threshold is evaluated to enhance iterative selections.

This process is repeated continuously for $\square_{\square}$ iterations, and finally a set of features is selected via Eq.15,

$$F_{final} = \bigcup_{i=1}^{f>2f_{th}} Feat_i \tag{15}$$

where, $Feat_i$ are the features selected by the i^{th} bacteria solutions.

4.4 Ensemble Classification

The features identified through BFO are used to identify different forensic events via use of an ensemble learning model, which combines Naïve Bayes (NB), k Nearest Neighbors (kNN), Support Vector Machine (SVM), Multilayer Perceptron (MLP), and Deep Forest (DF) set of classifiers. These

classifiers are reconfigured as per their hyperparameters mentioned in Table 2, which assists in enhancing their accuracy of forensic investigations.

Table 2. The ensemble learning classifier parameters for high efficiency levels.

Classifier Used for Forensic Investigations	Parameter Sets for these classifiers
Naïve Bayes (NB)	Level of Priors $= \frac{\sum_{j=1}^{n}(f_j - \frac{\sum_{a=1}^{n} f_a}{n})^2}{n-1}$ (16) where, n are total number of features, while f are the values of these features. Smoothing Constant $= L_r$
Logistic Regression (LR)	Use Normalized Samples = True Tolerance $= \frac{L_r}{N_c}$ (17) where, N_c are total number of forensic event classes. Maximum Iterations (MI) $= N_c * N_i$ (18)
Multilayer Perceptron (MLP)	Total Hidden Layers $= N_c$ Used Solver = SGD (Stochastic Solver with Gradient Descent) Rate of neuron learning $= L_r$
SVM	Coefficient of regularization $= L_r$ Used Kernel = Tan Sigmoid Weights of class $= (f_j - \frac{\sum_{a=1}^{n} f_a}{n})^2$ (19) where, j is the class number used for evaluations
DF	Total Trees $= N_c * NB$ Depth of the Forest $= N_i * N_c$

The values of these parameters are highly dynamic, and they are modified for each evaluation for improving accuracy of forensic analysis. The classes obtained by these classifiers are fused by a boosting process, which is done via Eq.20, as follows,

$$c_{out} = C(NB) * A(NB) + C(LR) * A(LR) + C(MLP) * A(MLP) + C(SVM) * A(SVM) + C(DF) * A(DF) \quad (20)$$

where, C & A represents the forensic event class, and accuracy of classification for the given classifier under the given forensic events. This fusion of classifiers assists in obtaining high-efficiency classifications with kernel-level security due to integration of blockchains.

Performance of this model in terms of accuracy of investigation, precision, recall & delay needed for investigations along with IoT network's QoS parameters can be observed from the next section.

5. EXPERIMENT AND ANALYSIS

The proposed model discusses design of a scalable kernel-component that is compatible with the majority of modern IoT kernels and aids in the

efficient pattern analysis of collected data samples. The proposed kernel component includes a Temporal Blockchain Cache (TBC) that is refreshed after analyzing each evidence set. The model proposes using a multidomain feature extraction engine that combines lightweight Fourier, Wavelet, Convolutional, Gabor, and Cosine feature sets to perform this analysis. A stochastic Bacterial Foraging Optimizer (BFO) selects these feature sets, which aids in the identification of high variance features. The selected features are then processed by an ensemble learning (EL) classifier comprised of the Nave Bayes (NB), k Nearest Neighbors (kNN), Logical Regression (LR), and Multilayer Perceptron (MLP) classifier sets.

To validate the security performance this model was tested on CSafe Forensics Data Samples, NIST Special Database, Computer Forensic Reference Dataset Samples, Breitinger Data Samples. All these data samples are publicly available, and were combined to identify network intrusions, packet-level attacks, and normal data flows.

Based on this strategy, the PoT blockchain's performance was estimated in terms of storage delay (d), energy needed for storage (E), throughput (T) and miner PDR during mining operations. These parameters were estimated via equations 21, 22, 23 & 24 and compared with FHB [2], HNF [8], and FL NTF [28] in Table 3.

$$d = \frac{1}{N_t}\sum_{i=1}^{N_t} t_{complete_i} - t_{start_i} \tag{21}$$

where, $t_{complete}$ & t_{start} represents the timestamps for completing and starting the evaluation process, while N_t are total number of events used during these evaluations.

$$E = \frac{1}{N_t}\sum_{i=1}^{N_t} e_{start_i} - e_{complete_i} \tag{22}$$

where, e represents the energy needed by miner nodes during mining operations.

$$THR = \frac{1}{N_t}\sum_{i=1}^{N_t} \frac{B_{rx_i}}{d_i} \tag{23}$$

where, B_{rx} represents total number of currently mined blocks without errors.

$$PDR = \frac{1}{N_t}\sum_{i=1}^{N_T} \frac{B_{rx_i}}{B_{tx_i}} \tag{24}$$

where, B_{tx} represents total number of blocks transmitted during the mining process.

Table 3. QoS performance under different network scenarios

Params. Used	FHB [2]	HNF [8]	FL NTF [28]	KFR EIAN
d (us)	2101	3728	2619	2030
E (mJ)	23.9	24.3	29.8	19.4
THR (kbps)	232	159	213	247
PDR (%)	89.1	75.7	87.8	98.5

To validate the event classification performance, the model was evaluated in terms of accuracy (A), precision (P), and recall (R) levels, which were estimated via equations 25, 26 and 27 as follows,

$$A = \frac{1}{N_c}\sum_{i=1}^{N_c}\frac{t_{p_i} + t_{n_i}}{t_{p_i} + t_{n_i} + f_{p_i} + f_{n_i}} \tag{25}$$

where, N_c are total number of event classes for which the model was evaluated, while t & f represents standard true & false rate sets.

$$P = \frac{1}{N_c}\sum_{i=1}^{N_c}\frac{t_{p_i}}{t_{p_i} + f_{p_i}} \tag{26}$$

$$R = \frac{1}{N_c}\sum_{i=1}^{N_c}\frac{t_{p_i}}{t_{p_i} + f_{n_i}} \tag{27}$$

The positive rates indicate classification of events in correct categories, while negative rates indicate their classification into incorrect categories. As per the similar strategy opted for identification of QoS performance, the accuracy performance measures were evaluated w.r.t. different number of forensic samples (NFS) as shown in Table 4.

The proposed forensic event classification model uses BFO for identification of highly variant feature sets, which assists the model to improve its forensic classification performance under different event types. It can be observed that the proposed model is able to improve the forensic classification precision by 0.5% when compared with FHB [2], 0.4% when compared with HNF [8], and 8.5% when compared with FL NTF [28] under different use cases. This precision is a measure of consistency and is improved due to use of BFO for identification of highly consistent feature sets, that assists in enhancing event detection performance for different use cases. Similarly, the recall of classification of these events can be observed from Table 5. It can be observed that the proposed model is able to improve recall of forensic event classification by 8.3% when compared with FHB [2], 4.9% when compared

with HNF [8], and 2.5% when compared with FL NTF [28] under different use cases. The reason for this enhancement is use of low-complexity feature sets, and their classification via ensemble classification process. The delay needed for this model is depicted in Figure 2.

Table 4. Forensic event classification accuracy for different models

NFS	Acc. FHB [2]	Acc. HNF [8]	Acc. FL NTF [28]	Acc. KFR EAIN
116k	83.44	88.78	84.96	93.12
174k	83.57	89.12	85.19	93.37
232k	83.68	89.45	85.40	93.60
290k	83.79	89.78	85.61	93.83
350k	83.92	90.11	85.83	94.07
412k	84.05	90.45	86.06	94.31
475k	84.20	90.78	86.30	94.57
534k	84.36	91.11	86.54	94.82
582k	84.53	91.45	86.78	95.08
640k	84.69	91.80	87.04	95.35
708k	84.86	92.16	87.30	95.63
767k	85.01	92.51	87.54	95.89
790k	85.15	92.86	87.77	96.15
873k	85.29	93.20	88.01	96.40
941k	85.43	93.55	88.24	96.66
1M	85.57	93.88	88.47	96.90

Table 5. Forensic event classification recall for different models

NFS	Rec. FHB [2]	Rec. HNF [8]	Rec. FL NTF [28]	Rec. KFR EAIN
116k	80.09	80.62	83.63	87.25
174k	80.30	80.99	83.93	87.53
232k	80.47	81.35	84.22	87.81
290k	80.61	81.71	84.51	88.09
350k	80.76	82.07	84.82	88.38
412k	80.92	82.44	85.13	88.68
475k	81.11	82.82	85.45	88.99
534k	81.33	83.21	85.76	89.30
582k	81.58	83.62	86.07	89.63
640k	81.85	84.05	86.38	89.96
708k	82.14	84.52	86.70	90.32
767k	82.38	84.94	87.00	90.64
790k	82.61	85.35	87.30	90.95
873k	82.83	85.76	87.61	91.27
941k	83.02	86.15	87.92	91.58
1M	83.20	86.52	88.22	91.87

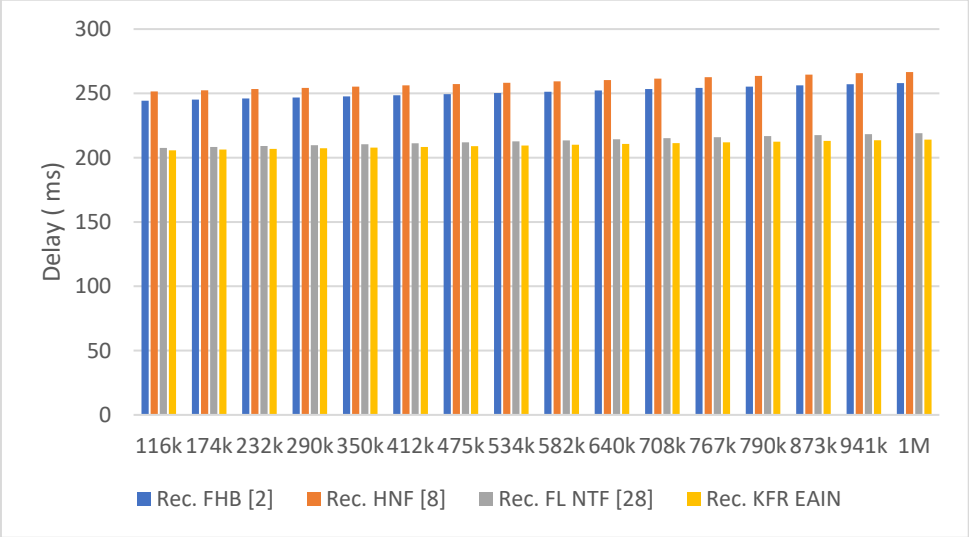

Figure 2. Forensic event classification delay for different models

Based on this evaluation and its visualization in Figure 2, it can be observed that the proposed model is able to improve speed of forensic event classification by 9.5% when compared with FHB [2], 12.4% when compared with HNF [8], and 0.5% when compared with FL NTF [28] under different use cases. The reason for this enhancement is use of low-complexity feature sets, and their classification via ensemble classification process. Due to these optimizations, the proposed model is highly useful for a wide variety of forensic investigation use cases.

The improvement in mining speed is attributed to the use of Proof-of-Time (PoT) based consensus mechanism in the proposed model. This consensus mechanism incorporates temporal delay metrics during mining operations, which allows for more efficient and faster processing of data. As a result, the delay in mining operations is significantly reduced, leading to improved mining speed and overall system performance. The use of PoT-based consensus is a key feature of the proposed model, and it has demonstrated its effectiveness in improving the efficiency and speed of mining operations. The improvement in forensic classification accuracy is attributed to the use of boosted ensemble classification in the proposed model. Boosted ensemble classification is a machine learning technique that combines the output of multiple classifiers to improve classification performance. In the proposed model, this technique is used to enhance event detection performance for different use cases. By combining the results of multiple classifiers, the proposed model is able to accurately classify events under different scenarios, leading to improved forensic classification accuracy.

6. CONCLUSION

The proposed model is able to improve its forensic classification performance across a wide variety of event types because it makes use of multidomain feature fusion and combines the results of those fusions with BFO

and ensemble learning classifiers. It was found that the proposed model has the potential to improve the accuracy of forensic classification by 10.5% in comparison to FHB [2], 2.9% in comparison to HNF [8], and 5.9% in comparison to FL NTF [28] in a variety of use cases. This accuracy is improved through the utilisation of boosted ensemble classification, which contributes to the improvement of the overall performance of event detection across a variety of use cases. The BFO algorithm is used in the proposed forensic event classification model for the purpose of identifying highly variant feature sets. This helps to improve the model's classification performance for a wide variety of event types. In terms of classification consistency, the proposed model is able to improve forensic classification precision by 0.5% when compared to FHB [2], 0.4% when compared to HNF [8], and 8.5% when compared to FL NTF [28] for a variety of use cases. These improvements were achieved by comparing the proposed model to FHB [2], HNF [8], and FL NTF [28]. This precision is a measure of consistency, and it is improved by the use of BFO to identify highly consistent feature sets, which in turn improves the performance of event detection for a variety of different use cases. In terms of scalability, it was found that the proposed model can improve the recall of forensic event classification by 8.3% when compared to FHB [2], 4.9% when compared to HNF [8, and 2.5% when compared to FL NTF [28] for a variety of use cases. These percentages were determined by comparing the proposed model to FHB [2], HNF [8], and FL NTF [28] respectively. The utilisation of low-complexity feature sets and the classification of those sets through the application of an ensemble classification process were the primary contributors to this improvement. As a result of these enhancements, the model that is being proposed is highly applicable to an extensive variety of use cases involving forensic investigations. In future, performance of this model may be validated on different kernel deployments, and can be enhanced via use of hybrid bioinspired models that allow for low-complexity feature selection and improved classification under real-time scenarios. This performance can also be improved via use of reinforcement learning with high-density feature extraction using a fusion of Long-Short-Term Memory (LSTM) with Gated Recurrent Units (GRU), and use of Auto Encoders (AE) with Q-Learning operations. This will allow the model to enhance its performance for a wide variety of real-time on-field deployments.

REFERENCES

[1] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis and E. K. Markakis, **A Survey on the Internet of Things (IoT) Forensics: Challenges, Approaches, and Open Issues**, *in IEEE Communications Surveys & Tutorials,* Vol. 22, No. 2, pp. 1191-1221, Second quarter 2020.

[2] W. A. Mahrous, M. Farouk and S. M. Darwish, **An Enhanced Blockchain-Based IoT Digital Forensics Architecture Using Fuzzy Hash**, *in IEEE Access,* Vol. 9, pp. 151327-151336, 2021.

[3] A. P. Sayakkara and N. -A. Le-Khac, **Electromagnetic Side-Channel Analysis for IoT Forensics: Challenges, Framework, and Datasets**, *in IEEE Access*, Vol. 9, pp. 113585-113598, 2021.
[4] J. Hou, Y. Li, J. Yu and W. Shi, **A Survey on Digital Forensics in Internet of Things**, *in IEEE Internet of Things Journal*, Vol. 7, No. 1, pp. 1-15, Jan. 2020.
[5] A. Al-Dhaqm et al., **Digital Forensics Subdomains: The State of the Art and Future Directions**, *in IEEE Access*, Vol. 9, pp. 152476-152502, 2021.
[6] D. Kim, Y. Pan and J. H. Park, **A Study on the Digital Forensic Investigation Method of Clever Malware in IoT Devices**, *in IEEE Access*, Vol. 8, pp. 224487-224499, 2020.
[7] Z. Li, H. Ren, E. Chou, X. Liu and C. D. McAllister, **Retrieving Forensically Sound Evidence from the ESP Series of IoT Devices**, *in IEEE Internet of Things Journal*, Vol. 9, No. 15, pp. 13144-13152, 1 Aug.1, 2022.
[8] Z. Liao, X. Pang, J. Zhang, B. Xiong and J. Wang, **Blockchain on Security and Forensics Management in Edge Computing for IoT: A Comprehensive Survey**, *in IEEE Transactions on Network and Service Management*, Vol. 19, No. 2, pp. 1159-1175, June 2022.
[9] L. Zhou, Y. Hu and Y. Makris, **A Hardware-Based Architecture-Neutral Framework for Real-Time IoT Workload Forensics**, *in IEEE Transactions on Computers,* Vol. 69, No. 11, pp. 1668-1680, 1 Nov. 2020.
[10] R. Zhao et al., **A Novel Intrusion Detection Method Based on Lightweight Neural Network for Internet of Things**, *in IEEE Internet of Things Journal*, Vol. 9, No. 12, pp. 9960-9972, 15 June15, 2022.
[11] G. Parise, D. Mohla, L. Parise and M. Lombardi, **IoT Innovations and Forensic Engineering in the Digital Age**, *in IEEE Transactions on Industry Applications*, Vol. 57, No. 3, pp. 2098-2103, May-June 2021.
[12] J. Cui, X. Zhang, H. Zhong, J. Zhang and L. Liu, **Extensible Conditional Privacy Protection Authentication Scheme for Secure Vehicular Networks in a Multi-Cloud Environment**, *in IEEE Transactions on Information Forensics and Security*, Vol. 15, pp. 1654-1667, 2020.
[13] L. Wei, J. Cui, Y. Xu, J. Cheng and H. Zhong, **Secure and Lightweight Conditional Privacy-Preserving Authentication for Securing Traffic Emergency Messages in VANETs**, *in IEEE Transactions on Information Forensics and Security*, Vol. 16, pp. 1681-1695, 2021.
[14] J. Zhang, H. Zhong, J. Cui, Y. Xu and L. Liu, **SMAKA: Secure Many-to-Many Authentication and Key Agreement Scheme for Vehicular Networks**, *in IEEE Transactions on Information Forensics and Security*, Vol. 16, pp. 1810-1824, 2021.
[15] X. Zhang, H. Zhong, C. Fan, I. Bolodurina and J. Cui, **CBACS: A Privacy-Preserving and Efficient Cache-Based Access Control Scheme for Software Defined Vehicular Networks**, *in IEEE Transactions on Information Forensics and Security*, Vol. 17, pp. 1930-1945, 2022.
[16] Q. Zhang, J. Wu, H. Zhong, D. He and J. Cui, **Efficient Anonymous Authentication Based on Physically Unclonable Function in**

Industrial Internet of Things, *in IEEE Transactions on Information Forensics and Security*, Vol. 18, pp. 233-247, 2023.

[17] Z. Abdullah, G. Chen, M. A. M. Abdullah and J. A. Chambers, **Enhanced Secrecy Performance of Multihop IoT Networks with Cooperative Hybrid-Duplex Jamming**, *in IEEE Transactions on Information Forensics and Security*, Vol. 16, pp. 161-172, 2021.

[18] S. Rajendran and Z. Sun, **RF Impairment Model-Based IoT Physical-Layer Identification for Enhanced Domain Generalization**, *in IEEE Transactions on Information Forensics and Security*, Vol. 17, pp. 1285-1299, 2022.

[19] N. Y. Ahn and D. H. Lee, **Security of IoT Device: Perspective Forensic/Anti-Forensic Issues on Invalid Area of NAND Flash Memory**, *in IEEE Access*, Vol. 10, pp. 74207-74219, 2022.

[20] N. V. Abhishek, A. Tandon, T. J. Lim and B. Sikdar, **A GLRT-Based Mechanism for Detecting Relay Misbehavior in Clustered IoT Networks**, *in IEEE Transactions on Information Forensics and Security*, Vol. 15, pp. 435-446, 2020.

[21] F. Tong, X. Chen, K. Wang and Y. Zhang, **CCAP: A Complete Cross-Domain Authentication Based on Blockchain for Internet of Things**, *in IEEE Transactions on Information Forensics and Security*, Vol. 17, pp. 3789-3800, 2022.

[22] S. Zhao, S. Li, L. Qi and L. D. Xu, **Computational Intelligence Enabled Cybersecurity for the Internet of Things**, *in IEEE Transactions on Emerging Topics in Computational Intelligence*, Vol. 4, No. 5, pp. 666-674, Oct. 2020.

[23] Y. Zhou, G. Cheng and S. Yu, **An SDN-Enabled Proactive Defense Framework for DDoS Mitigation in IoT Networks**, *in IEEE Transactions on Information Forensics and Security*, Vol. 16, pp. 5366-5380, 2021.

[24] L. David, A. Hassidim, Y. Matias, M. Yung and A. Ziv, **Eddystone-EID: Secure and Private Infrastructural Protocol for BLE Beacons**, *in IEEE Transactions on Information Forensics and Security*, Vol. 17, pp. 3877-3889, 2022.

[25] H. M. J. Almohri, L. T. Watson and D. Evans, **An Attack-Resilient Architecture for the Internet of Things**, *in IEEE Transactions on Information Forensics and Security*, Vol. 15, pp. 3940-3954, 2020.

[26] A. Vangala, A. K. Das, A. Mitra, S. K. Das and Y. Park, **Blockchain-Enabled Authenticated Key Agreement Scheme for Mobile Vehicles-Assisted Precision Agricultural IoT Networks**, *in IEEE Transactions on Information Forensics and Security*, Vol. 18, pp. 904-919, 2023.

[27] B. Ahuja, D. Mishra and R. Bose, **Fair Subcarrier Allocation for Securing OFDMA in IoT Against Full-Duplex Hybrid Attacker**, *in IEEE Transactions on Information Forensics and Security*, Vol. 16, pp. 2898-2911, 2021.

[28] P. Zhang, Y. Tao, Q. Zhao and M. Zhou, **A Rate-and-Trust-Based Node Selection Model for Block Transmission in Blockchain Networks**, *in IEEE Internet of Things Journal*, Vol. 10, No. 2, pp. 1605-1616, 15 Jan.15, 2023.
[29] Z. He et al., **Edge Device Identification Based on Federated Learning and Network Traffic Feature Engineering**, *in IEEE Transactions on Cognitive Communications and Networking*, Vol. 8, No. 4, pp. 1898-1909, Dec. 2022.
[30] L. Li, Y. Luo, J. Yang and L. Pu, **Reinforcement Learning Enabled Intelligent Energy Attack in Green IoT Networks**, *in IEEE Transactions on Information Forensics and Security*, Vol. 17, pp. 644-658, 2022.
[31] M. I. Ali et al., **Security Challenges and Cyber Forensic Ecosystem in IoT Driven BYOD Environment**, *in IEEE Access*, Vol. 8, pp. 172770-172782, 2020.
[32] Q. Luo, J. Liu, J. Wang, Y. Tan, Y. Cao and N. Kato, **Automatic Content Inspection and Forensics for Children Android Apps**, *in IEEE Internet of Things Journal*, Vol. 7, No. 8, pp. 7123-7134, Aug. 2020.
[33] T. Trajanovski and N. Zhang, **An Automated and Comprehensive Framework for IoT Botnet Detection and Analysis (IoT-BDA)**, *in IEEE Access*, Vol. 9, pp. 124360-124383, 2021.
[34] A. Nieto, **Becoming JUDAS: Correlating Users and Devices During a Digital Investigation**, *in IEEE Transactions on Information Forensics and Security*, Vol. 15, pp. 3325-3334, 2020.
[35] Z. Zhou et al., **Coverless Information Hiding Based on Probability Graph Learning for Secure Communication in IoT Environment**, *in IEEE Internet of Things Journal*, Vol. 9, No. 12, pp. 9332-9341, 15 June15, 2022.
[36] G. Xu et al., **An Ensemble Learning-Based Prediction Model for Image Forensics From IoT Camera in Smart Cities**, *in IEEE Access*, Vol. 8, pp. 222117-222125, 2020.
[37] M. R. Nosouhi, K. Sood, M. Grobler and R. Doss, **Towards Spoofing Resistant Next Generation IoT Networks**, *in IEEE Transactions on Information Forensics and Security*, Vol. 17, pp. 1669-1683, 2022.
[38] L. Sun, Y. Wang, Z. Qu and N. N. Xiong, **BeatClass: A Sustainable ECG Classification System in IoT-Based eHealth**, *in IEEE Internet of Things Journal*, Vol. 9, No. 10, pp. 7178-7195, 15 May15, 2022.
[39] E. Dushku, M. M. Rabbani, M. Conti, L. V. Mancini and S. Ranise, **SARA: Secure Asynchronous Remote Attestation for IoT Systems**, *in IEEE Transactions on Information Forensics and Security*, Vol. 15, pp. 3123-3136, 2020.
[40] X. Xu, X. Liu, Z. Xu, F. Dai, X. Zhang and L. Qi, **Trust-Oriented IoT Service Placement for Smart Cities in Edge Computing**, *in IEEE Internet of Things Journal*, Vol. 7, No. 5, pp. 4084-4091, May 2020.

[41] S. Yılmaz, E. Aydogan and S. Sen, **A Transfer Learning Approach for Securing Resource-Constrained IoT Devices**, *in IEEE Transactions on Information Forensics and Security*, Vol. 16, pp. 4405-4418, 2021.

[42] N. Wang, W. Li, A. Alipour-Fanid, L. Jiao, M. Dabaghchian and K. Zeng, **Pilot Contamination Attack Detection for 5G MmWave Grant-Free IoT Networks**, *in IEEE Transactions on Information Forensics and Security*, Vol. 16, pp. 658-670, 2021.

[43] S. Ma, Y. Zhong and Q. Huang, **Efficient Public Key Encryption With Outsourced Equality Test for Cloud-Based IoT Environments**, *in IEEE Transactions on Information Forensics and Security*, Vol. 17, pp. 3758-3772, 2022.

[44] I. Ali et al., **Systematic Literature Review on IoT-Based Botnet Attack**, *in IEEE Access*, Vol. 8, pp. 212220-212232, 2020.

[45] Z. Jin, C. Zhang, Y. Jin, L. Zhang and J. Su, **A Resource Allocation Scheme for Joint Optimizing Energy Consumption and Delay in Collaborative Edge Computing-Based Industrial IoT**, *in IEEE Transactions on Industrial Informatics*, Vol. 18, No. 9, pp. 6236-6243, Sept. 2022.

[46] Y. Yu and J. Liu, **TAPInspector: Safety and Liveness Verification of Concurrent Trigger-Action IoT Systems**, *in IEEE Transactions on Information Forensics and Security*, Vol. 17, pp. 3773-3788, 2022.

[47] N. Singhal, V. Ganganwar, M. Yadav, A. Chauhan, M. Jakhar, and K. Sharma, **Comparative Study of Machine Learning and Deep Learning Algorithm for Face Recognition**, *In Jordanian Journal of Computers and Information Technology* (Issue 0, p. 1), 2021.

[48] A. Angbera, and H. Chan, **A Novel True Real-Time Spatiotemporal Data Stream Processing Framework**, *in Jordanian Journal of Computers and Information Technology* (Issue 0, p. 1), 2022.

[49] N. Yassin, **Data Hiding Technique for Color Images using Pixel Value Differencing and Chaotic Map**, *In Jordanian Journal of Computers and Information Technology* (Issue 0, p. 1), 2022.

[50] Z. Ashi, L. Aburashed, M. Qudah, and A. Qusef, **Network Intrusion Detection Systems Using Supervised Machine Learning Classification and Dimensionality Reduction Techniques: A Systematic Review**, *In Jordanian Journal of Computers and Information Technology* (Issue 0, p. 1), 2021.

[51] A. Oussous, and F. Benjelloun, **A Comparative Study of Different Search and Indexing Tools for Big Data**, *In Jordanian Journal of Computers and Information Technology* (Issue 0, p. 1), 2022.

[52] I. Almomani, and K. Sundus, **The Impact of Mobility Models on the Performance of Authentication Services in Wireless Sensor Networks**, *In Jordanian Journal of Computers and Information Technology* (Issue 0, p. 1), 2020.