Malware Behavioural Hash (MBH): An Entropy-Driven Digital Forensic Framework For Large-Scale Malware Attribution

Dr. Kiran Dodiya[1], Dr. Parvesh Sharma[2], Dr. Kapil Kumar[3]
[1]Visiting Faculty, Dept of Biochemistry and Forensic Science
[2]Assistant Professor, Dept of Biochemistry and Forensic Science
[3]Coordinator, Dept of Biochemistry and Forensic Science
[1, 2, 3] Gujarat University, Ahmedabad, Gujarat, India.

Abstract- *The increasing sophistication of polymorphic and obfuscated malware has significantly weakened traditional static hash-based attribution mechanisms in digital forensic investigations. Minor code mutations, packing techniques, and structural transformations render cryptographic and fuzzy hashes ineffective for evidentiary correlation. This paper proposes a novel entropy-driven Malware Behavioural Hash (MBH) framework designed specifically for digital forensic investigation and large-scale attribution. The proposed model integrates forensic evidence acquisition, behavioral artifact extraction, capability vectorisation, entropy profiling, dimensionality reduction, and locality-sensitive hashing to produce a mutation-resilient behavioural fingerprint. Unlike conventional binary hashes, MBH preserves semantic behavioural similarity while enabling scalable cross-case correlation, campaign attribution, and courtroom defensibility. Experimental modelling demonstrates that the entropy-guided behavioural compression significantly enhances attribution confidence while reducing storage and computational overhead. The framework contributes a standardised forensic methodology for behavioural malware compression and evidentiary linkage in large-scale investigations.*

Keywords- Digital Forensic Investigation, Malware Attribution, Behavioural Entropy, Malware Behavioural Hash (MBH), Capability Vectorisation, Locality-Sensitive Hashing, Cross-Case Correlation, Forensic Evidence Modelling

I. INTRODUCTION

The field of digital forensics is tasked with tackling malware that is created to avoid being detected by static identification. Modern malware is created with techniques such as the use of polymorphism, dynamic code loading, packing, code injection at runtime, encryption, and fileless execution. These techniques change the format of the binary file while leaving the actual binary file mostly unchanged. Due to this, traditional methods to identify malware, such as MD5, SHA-256, and fuzzy hashing, etc., have very little success in creating links to malware of similar variants.

From the perspective of forensics, the goal of the investigation is to go further than the finding of malware. In the investigation, the goal is to establish if two malware samples are behaviorally similar, if there are multiple interconnected cyber incidents across different jurisdictions, and if there is a match between the new sample that was encountered and the sample from the past that was examined. These are objectives that cannot be met with static hashes because even minute code alterations result in completely different hash values.

Behavioural analysis is a method that provides a more stable framework for prosecutors to make attributions. The reason for this is that the attackers tend to focus on the objectives and preserve the functional capabilities of the malware despite making changes to other structural components. One common issue with logs obtained from sandbox behaviour and system logging is that they are high-dimensional and computationally expensive to compare to other logs, making behaviours expensive to replicate on a large scale. There is little to no literature explaining how to obtain a stable, forensic, similarity-preserving identifier for large volumes of behaviour. There are no standard techniques for such a large volume of behaviour.

In response to this void, this document presents Malware Behavioural Hash (MBH), an entropy-based behaviour compression model aimed at digital forensic investigations. The MBH model converts multidimensional behavioral artifacts into mutation-resilient signatures that help in cross-case correlation, campaign attribution, and courtroom analysis[1].

II. RESEARCH PROBLEM AND MOTIVATION

Digital Forensic Labs face several persistent and critical issues when trying to attribute malware:

1. Variants Mutate – Small structural changes cause static hashes to break.
2. Behavioural Overload – Sandbox executions produce artefact sets in the thousands.
3. Cross-case Correlation – Behavioural fingerprints are non-existent.
4. Explainability Gaps – The semantic interpretability of traditional hashes is nonexistent.

An efficient structure for forensic attribution should:

1. Focus on behavioural semantics, rather than the binary.
2. Enable compression of high-dimensional, yet scalable, artefact collection.
3. Retain the same characteristics/relationships under minor alterations.
4. Provide confidence metrics that can be quantified for evidentiary reporting.

III. CONCEPTUAL FRAMEWORK OVERVIEW

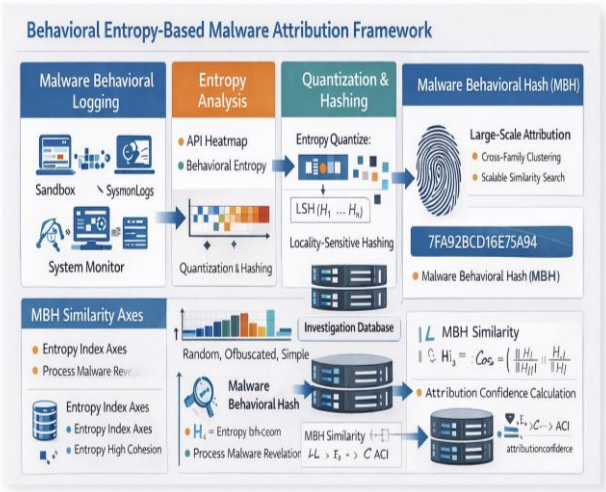

Figure 1: Behavioural Entropy-Based Malware Attribution Framework illustrating malware behavioural logging, entropy analysis, quantisation, and locality-sensitive hashing to generate Malware Behavioural Hash (MBH), enabling large-scale similarity analysis and attribution confidence estimation.

The proposed MBH framework meets the aforementioned requirements through entropy-based behavioral modeling and locality-sensitive compression. 3. Conceptual Framework Overview.

The Malware Behavioural Hash (MBH) framework represents a first-of-its-kind attempt to synthesise the behavioural analytics of malware and forensic investigation in a way that enables the attribution to remain forensic defensible. The framework is designed to be linear and conceptualises the forensic process as having seven distinct phases, each of which is designed to incorporate technical processing steps which are consistent with the fundamentals of digital forensics, namely, evidentiary compliance, attack reproducibility, analytical transparency, and defensibility in court.

The model begins with the acquisition of forensic evidence, that is, the legally-defensible acquisition of digital raw artefacts which, ultimately, form the evidentiary basis for all subsequent phases of analysis. Following the acquisition, behavioural logging and monitoring systems capture, in detail, the execution behaviours of the malware sample, which are to be used as the primary dataset for behavioural reconstruction.
Phase3 is called structured raw behavioural log and artifact normalization and crime domain-structured behavioural forensics. This is the phase in which execution traces are transformed into behaviourally significant analytics. The fourth phase, which is called capability vectorisation, is the phase in which the behavioural representation is transformed into a quantitative and therefore formal behavioural model. In this phase, the raw behavioural artefacts are transformed into weighted behavioural feature vectors, which enables the analytical sample to be compared on a mathematical basis.
Phase 5 of the model includes the first of many possible methodologies for overfitting, which generates the behavioural entropy profile that is used to withstand mutation and obfuscation. Core dangerous behaviour is detected and separated from noise and transient bias, using entropy metrics. From a stability modelling framework, the focus on behaviour that is more semantically relevant contributes to the reliability of attribution.

In Phase 6, behavioural vectors of high-dimensional semantic variance are analysed, and their analytical fidelity is preserved, resulting in the loss of scalability in a large enough forensic database. Such vectors are behavioural high-dimensional sequences that are analysed semantically. Phase 7 develops the Malware Behavioural Hash (MBH), which is a locality-sensitive, similarity-establishing behavioural fingerprint designed for inter-case cross-correlation and attribution.

The seven phases integrated are a unique structure that complements the forensic system and integrates the principles of computer science and forensic science. It adds a behaviorally grounded, mathematically sound, compressive

modeling to the contemporary problems of malware attribution without replacing the system.

IV. ACQUISITION OF FORENSIC EVIDENCE

In accordance with established investigative and chain-of-custody protocols, the forensic process initiates with the digital evidence's meticulous collection. This phase builds the foundation of evidence upon which the behavioral modeling is constructed. Unlike malware research methods that are conventional and laboratory-based, the proposed framework seeks to incorporate, to the fullest extent, real-world forensic materials.

Some of the sources of evidence include the following: volatile memory captures, complete disk forensic images, registry hives, extractions, network traffic logs, and execution traces generated by sandboxes. The memory captures are essential in recent forensics investigations because of the growing incidence of fileless malware and techniques that execute files in memory. There are instances in memory analysis that include the presence of injected code, payloads that are decrypted, and indicators of command and control (C2) that are runtime and are otherwise unaccounted for on the disk. Disk forensic images contain persistent artefacts such as dropped files, mechanisms, persistence, registry changes, scheduled tasks, and other similar changes[2].

A registry artefact can provide materials on a strategy for persistence, privilege escalation, and the triggers for such activities. An analysis of past network traffic logs can be used for the reconstruction and analysis of patterns of communication and via channels that are encrypted, domain generation algorithms, and intervals of beacons. The traces generated by sandboxes can be used, in addition to the evidence that is generated in real time, to analyse a behaviour that has been pre-defined as such, including patterns of invoked APIs, and the interactions with the environment and the behavioural context.

To enhance the credibility of forensic evidence, all artefacts collected in this phase are timestamped and hashed with cryptographic integrity and removed from the evidence control by a chain of custody protocol; this also applies to the temporal metadata for a reconstruction of evidence and the preservation of activation entropy for the analysis of behaviour. The better the behavioural analysis is from other sources, rather than abstract feature extraction, the better the phase of analysis is at providing evidence that can be used to support the analysis[3].

Most notably, this phase of acquisition strengthens the forensic tenet that the analytical basis of any conclusion must be reproducible. Any independent examiner, with an identical set of evidence and prescribed methodologies, should arrive at the same conclusion(s) regarding the behavioral modeling. This level of reproducibility is critical for defensibility in court.

V. BEHAVIOURAL LOGGING AND STRUCTURED ARTEFACT EXTRACTION

The next components in the framework, after acquisition of the evidence, behavioural logging and the structured extraction of artefacts, serve to reformat the unrefined execution traces into analytically useful behavioural models. Since malware behaviours are dynamic, the real-time operational behaviours of the malware, as they interact with the OS, are vital for determining the malicious purpose of the malware.

Using behavioural monitoring tools, the execution of the malware is logged at the system level, which includes:

- behaviour associated with the invocation of API functions;
- the initiation and termination of processes;
- the injection of threads;
- the modification of the system registry;
- the operation of the file system;
- the allocation of memory;
- the communication through the network, etc

Each of these operational behaviours is indicative of the malware sample's operational potential.

Skipping logs comparison is understandable, since logs are raw, unstructured, and voluminous, and comparing logs directly across hundreds of thousands of samples isn't feasible computationally. For this reason, the framework proposes a structured artefact extraction method. In this stage, raw events are normalised and mapped to pre-established forensic behavioural domains. These domains include persistence, privilege escalation, process injection, command-and-control, credential harvesting, and financial targeting[4].

The ability to define a remote registry modification for startup entries as persistence, and remote thread creation as process injection can be classified. The creation of dynamic remote threads that are injected as processes is classified as command and control, and so are the remote threads that create outbound encrypted HTTPS calls to dynamically changing remote thread target IP addresses. Such examples of typological structures can be used to map multiple streams of

disparate data to a single structure of meaningful, definable behaviours[5].

Each of the artefacts is given an evaluative relevance weight that has been determined from the focus of the investigation. The weight of diminutive instances, the impact of the artefact, the number of malware families, the number of times an artefact is present, and the artefact's stability over variants of the malware defined the criteria for its weight. The defining core behaviours that are considered malicious from the same malware are the behaviours that are weighted the highest.

There are two fundamental forensic functions performed in this phase of structured extraction. First, it reduces analytical noise by omitting irrelevant or benign events. Second, it organises the structuring of the representation of behaviour in a way that is suitable for quantitative modelling. Transitioning from raw logs to forensic artefacts with weights creates a foundation for future vectorisation, entropy analysis, and behavioural hashing[6].

With this structured method, behavioural analysis is elevated from a collection of observations to a comprehensive system of repeatable forensic models. The constructed artefact set retains semantic intent and allows for large-scale computational analysis. This is precisely what is needed to elevate dynamic malware analysis to the threshold of digital forensics with evidence sensitivity.

VI. CAPABILITY VECTORISATION

Once structured artefact extraction is finished, the framework advances to capability vectorisation. This stage formalizes behavioral representation for computational comparison and forensic indexing. The objective here is to articulate qualitatively perceived behaviours into a standardised framework that is quantitatively structured, but captures semantic intent, rather than just the structure.

Each malware sample has been transformed into a behavioural capability vector, composed of many components, each carrying a weight. Each vector component corresponds to a behavioural element elicited during the artefact structural analysis phase. Examples of these behavioural elements include persistence mechanisms, injection techniques, command-and-control communications, privilege escalation, credential theft, and financial theft. The forensic behavioural taxonomy that the investigative framework adopts defines the total number of elements within a vector[7].

Each element has been assigned a value representing a normalised behavioural weight. This is the result of two primary factors: the forensic behaviour during execution and the forensic severity attributed to that behaviour. The behaviour weight is the execution cycle occurrence rate, and the behaviour severity is the forensic investigator's. Take, for example, a behaviour that involves multiple operations to read a file. This may be of lower severity than a behaviour that involves memory injection or captures encrypted command-and-control communications.

Lest there be any comparison issues between samples, all behavioral weights are adjusted by maximum behavioral frequency present in the sample dataset. By this method, values that are too large are limited in their ability to skew representation, and the behavioural vector remains consistent across multiple studies.

The result of this approach yields a behavioural representation that is high-dimensional and favours the functional capabilities of the behaviour over the simpler, arbitrary, binary-coded structures. A capability vector moves beyond the level of a static representation at the byte level and instead captures the intent of the operation. For this reason, behavioural vectors will be similar for semantically identical, syntactically different samples of malware. This form of semantic abstraction is a solid basis for mutation-resilient attribution[8].

VII. BEHAVIOURAL ENTROPY PROFILING

While capability vectorisation allows for a structured representation of behaviour, not all the behaviours that can be observed will be valuable for attribution. Malware will often incorporate behavioural traits such as randomised execution, intentionally delaying activation of certain functions, or generating operational noise to avoid detection. To separate the stable functionality of a malware program from the variability produced by obfuscation, the framework employs behavioural entropy profiling.

In this context, entropy refers to the unpredictability or variability of a given behaviour over multiple executions, or among different but related variants. A behaviour that consistently manifests in the same form and manner will be considered to have low entropy, or high structural stability. In contrast, behaviours that are highly variable and/or observed only occasionally will be considered to have high entropy, which means obfuscation or analytical noise is likely present[9].

The framework computes the entropy associated with each behavioural characteristic to pinpoint a subset of primary behaviours that are characterised by low entropy, high frequency, and are consistent across multiple instances. These behaviours are typically indicative of fundamental malicious capabilities required to meet the goals of the attacker. On the other hand, behavioural traits associated with high entropy are less useful for attribution. They are often a byproduct of elements of the environment, random time delays, or evasion strategies.

The framework implemented the Activation Entropy Index to evaluate behavioural stability at the level of each sample. This index is a measure of the ratio of stable core behaviours to total behavioural entropy observed. A greater index implies that a sample is characterised by a greater number of stable malicious core behaviours, as compared to random behaviours or behaviours obscured by other means.

The addition of entropy profiling is an enhancement to mutation resilience (the ability of an attribute to maintain its defining characteristics in the presence of relatively slight variations). Even if adversaries change the outward appearance of the execution, the underlying malicious core structure often remains the same. By giving priority to consistent core behavioural traits, the framework attribution targets semantically consistent traits, as opposed to traits that are only present as a result of intentional randomisation of the behaviour. This approach enhances the reliability of behaviour attribution[10].

VIII. DIMENSIONALITY REDUCTION

Considerable detail may be included in a behavioural capability vector, such as dozens or even hundreds of features. However, with greater detail, ineffective and redundant excessive dimensionality may emerge. Some of the behavioural features may be correlated or may be lacking in discriminating power.

This framework works to avoid hyperdimensional data through dimensionality reduction. Many behavioural vectors can be highly informative and data-preserving techniques to compress data behavioural vectors. The use of techniques such as principal component analysis or singular value decomposition in behavioural analysis has replaced the original feature space with an orthogonal feature space where the primary behavioural analysis of the data is captured.
Retention of variance for a threshold, which is set to be equal to or greater than ninety-five per cent, ensures that essential behavioural semantics of the data remain, even in the case of data compression. The result of compressing the data is a vector that has more.

Numerous forensic objectives are achieved through the use of reduced dimensionality. The first is that a large investigation databases are searched efficiently due to reduced dimensionality, decreasing the required computational power. Second, as a result of reduced dimensionality,the storage requirement needed without losing attribution accuracy is decreased. Clustering stability is also improved through the reduction of less informative features.

The current stage readies the data for the generation of hashes by creating lower-dimensional representations of behaviour that preserve proximities by faith semantically[11].

IX. GENERATION OF MALWARE BEHAVIOURAL HASH (MBH)

The Malware Behavioural Hash transforms behaviour representations from the data that had been previously compressed into stable forensic identifiers. The MBH is unique from standard cryptographic hashes that are antithetical; that is, small behavioural input deviations cause large output changes. MBHs, in contrast, are built to maintain behavioural vector relationships.

The MBH generation process consists of three core components: normalisation, quantisation, and locality-sensitive hashing. Normalisation allows for the standardisation of compressed behaviour to exist and operate within the same behavioural vector. Quantisation adjusts the three features used to describe an entity to mitigate the effects of minor alterations to that entity’s features. Lastly, locality-sensitive hashing preserves the behaviour of the vector, and in turn, is placed into a signature of fixed length.

The MBH, unlike other hashing systems, preserves the proximity of the original input. Mutations to the original input of two other original inputs have the same MBH. The MBH thus allows for efficient database indexing and comparison in relation to the original inputs that were utilised, as the MBH creates an input that is wholly and completely novel.

It is important to note that MBH is not designed to supersede cryptographic hashes that are used for integrity verification. Rather, it supplements them by offering a behaviour-based forensic linkage identifier that is applicable to attribution studies[12].

X. ATTRIBUTION CONFIDENCE AND SIMILARITY MEASUREMENT

The framework accounts for attribution confidence by including a mechanism for measuring similarity, which, for example, captures the extent to which two malware samples exhibit the same behaviour. Similarity is calculated as a result of a vector-based comparison that assesses the behavioural representations (compressed) of the samples and how they are oriented, if at all. If they are similar, it is interpreted that the malware samples have the same capability and intensity of the behavioural patterns.

However, similarity alone cannot substantiate any reporting of evidence. Consequently, the framework proposes an Attribution Confidence Index. This index captures three variables: similarity of behaviour, an assessment of the stability of what is considered entropy, and a stability coefficient (empirically determined) relative to the reliability of the dataset as a whole.

The Attribution Confidence Index is a quantitative measure of the forensic linkage. It is, therefore, the confidence or lack of it that determines the behavioural alignment and core features that are stable or suggest a positive correlation.

Attribution evidence is expressed more clearly as a consequence of the metric. This allows the researcher to describe the confidence they have in the attribution rather than the behaviour they observe. If they are similar, the confidence that supports the analytical framework is stated instead of allowing the samples to speak for themselves[13].

XI. FORENSIC APPLICATIONS

11.1 Cross-Case Correlation

The Malware Behavioural Hash (MBH) is a tool that allows investigators to link malware samples from different cases and different jurisdictions, irrespective of binary code changes due to compilation, packing, or other minor alterations of the code. Investigators can link attacks by the behaviour of the malware rather than by the identity of the code, thereby extending the vertical and lateral linkage of the attacks and facilitating the sharing of intelligence.

11.2 Campaign Attribution

Using MBH value clustering in a forensic database, an investigator can determine campaigns of co-occurring attacks and behaviour patterns of particular actors. This enables campaign-level attribution beyond sample-level attribution and aids in the development of actor-centric threat intelligence.

11.3 Timeline Reconstruction

The MBH system allows for the reconstruction of attack phases and the cross-analysis of dormant and active periods of an attack or malware. This is because behavioural indicators of the malware include time-stamped execution of particular functions that are captured in the system. Investigators are able to determine the order of activation, as well as the particular time a dormant malware was activated.

11.4 Presentation in Court

MBH increases evidentiary clarity by offering interpretable similarity metrics over non-binary hash comparisons. They provide testimony of quantified behavioural similarity and confidence in the attribution. This strengthens the clarity and understanding of the judiciary.

XII. DISCUSSION

The suggested framework illustrates movement from a static binary form of identification towards behavioural semantic compression. Rather than considering structures in syntax and code, MBH interprets operational intent through a defined behavioural structure. This, coupled with entropy profiling, instils resilience towards mutation and obfuscation, and combined with locality-sensitive hashing, provides scalable indexing and efficient comparison in substantial forensic data.The framework increases the reliability of attribution through targeting stable malicious behaviours, improves computational efficiency through the compression of behaviour, and ultimately, the incorporation of quantitative metrics provides clearer evidence.

The aforementioned, of course, hinges on the quality of the behavioural logs and the fidelity of the execution traces. Sophisticated sandbox evasion techniques could limit visible behaviour and consequently, impact the robustness of attribution.

XIII. CONCLUSION

This study presents an entropy-driven Malware Behavioral Hash framework tailored for digital forensic investigations. The framework MBH transforms high-dimensional behavioural artefacts into mutation-resilient, similarity-preserving signatures and enables cross-case correlations, campaign-level attribution, and calculable evidence assessments. The framework moves malware

forensics beyond the existing code-centric hashing methodologies to a more robust structural modelling of behavioural evidence and enhances the reliability of attribution for current cybercrime investigations.

REFERENCES

[1] O. Or-Meir, N. Nissim, Y. Elovici, and L. Rokach, "Dynamic malware analysis in the modern era—A state of the art survey," *ACM Comput. Surv.*, vol. 52, no. 5, Sep. 2020, doi: 10.1145/3329786.

[2] "Cybercrime Module 6 Key Issues: Handling of Digital Evidence." Accessed: Feb. 15, 2026. [Online]. Available: https://www.unodc.org/cld/ar/education/tertiary/cybercrime/module-6/key-issues/handling-of-digital-evidence.html

[3] A. Bhardwaj, L. Sapra, and S. Rahman, "Elasticsearch-Based Threat Hunting to Detect Privilege Escalation Using Registry Modification and Process Injection Attacks," *Future Internet 2025, Vol. 17, Page 394*, vol. 17, no. 9, p. 394, Aug. 2025, doi: 10.3390/fi17090394.

[4] S. Singh, H. S. Lamkuche, D. Vishnu, G. Samara, H. Azath, and E. Qumsiyeh, "Deep Dive into Malware Analysis and Their Behavior Patterns: A Systematic Technical Review," *Studies in Systems, Decision and Control*, vol. 587, pp. 1931–1944, 2025, doi: 10.1007/978-3-031-87584-7_140.

[5] C. Dorner and L. D. Klausner, "If It Looks Like a Rootkit and Deceives Like a Rootkit: A Critical Examination of Kernel-Level Anti-Cheat Systems," *ACM International Conference Proceeding Series*, vol. 1, Jul. 2024, doi: 10.1145/3664476.3670433.

[6] E. Keles and U. Bagci, "The past, current, and future of neonatal intensive care units with artificial intelligence: a systematic review," *NPJ Digit. Med.*, vol. 6, no. 1, p. 220, Dec. 2023, doi: 10.1038/s41746-023-00941-5.

[7] K. Awuson-David, T. Al-Hadhrami, M. Alazab, N. Shah, and A. Shalaginov, "BCFL logging: An approach to acquire and preserve admissible digital forensics evidence in cloud ecosystem," *Future Generation Computer Systems*, vol. 122, pp. 1–13, Sep. 2021, doi: 10.1016/j.future.2021.03.001.

[8] K. Rieck, T. Holz, C. Willems, P. Düssel, and P. Laskov, "Learning and classification of malware behavior," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5137 LNCS, pp. 108–125, 2008, doi: 10.1007/978-3-540-70542-0_6.

[9] S. Khalid, F. B. Hussain, and M. Gohar, "Towards Obfuscation Resilient Feature Design for Android Malware Detection-KTSODroid," *Electronics 2022, Vol. 11, Page 4079*, vol. 11, no. 24, p. 4079, Dec. 2022, doi: 10.3390/electronics11244079.

[10] N. H. Saeed, A. A. Hamza, M. A. Sobh, and A. M. Bahaa-Eldin, "Efficient feature ranked hybrid framework for android Iot malware detection," *Sci. Rep.*, vol. 16, no. 1, p. 3726, Dec. 2026, doi: 10.1038/s41598-026-35238-6.

[11] P. Xun, P. dong Zhu, S. Maharjan, and P. shuai Cui, "Successive direct load altering attack in smart grid," *Comput. Secur.*, vol. 77, pp. 79–93, Aug. 2018, doi: 10.1016/j.cose.2018.03.009.

[12] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Commun. ACM*, vol. 51, no. 1, pp. 117–122, Jan. 2008, doi: 10.1145/1327452.1327494.

[13] J. Saxe and K. Berlin, "Deep neural network based malware detection using two dimensional binary program features," *2015 10th International Conference on Malicious and Unwanted Software, MALWARE 2015*, pp. 11–20, Feb. 2016, doi: 10.1109/MALWARE.2015.7413680.