Veritas in caritate

Księga pamiątkowa ku czci Księdza Profesora Andrzeja Szostka MIC

Redakcja
o. Marcin Tkaczyk
Marzena Krupa
ks. Krzysztof Jaworski

Wydawnictwo KUL
Lublin 2016

Paweł Urbanowicz, Ryszard Kozera, Michał Dolecki

Zastosowanie sztucznych sieci neuronowych do uzgadniania kluczy kryptograficznych

Wprowadzenie

Wśród prowadzonych obecnie badań szczególnego znaczenia nabierają prace interdyscyplinarne. Również w informatyce często pojawiają się interesujące odkrycia z obszaru wzajemnego przenikania różnych specjalności. Tak też na styku sztucznej inteligencji i kryptologii dostrzeżono możliwość wykorzystania sztucznych sieci neuronowych jako mechanizmu ustalania wartości liczbowych niezbędnych do szyfrowania danych. Opublikowane przez Kinzela i Kantera prace[1] [2] zaprezentowały nowe zjawisko polegające na ustaleniu zgodnych wartości wag w sieciach o specyficznej budowie. Liczby te mogą być następnie wykorzystane podczas szyfrowania komunikacji między nadawcą i odbiorcą jako stosowne klucze kryptograficzne. Zaproponowany przez odkrywców tego zjawiska algorytm wzajemnego uczenia sieci jest interesującym obiektem dalszych badań ze względu na złożony, losowy charakter oraz ważne zastosowania praktyczne.

Człowiek wykształcił wiele różnorodnych form komunikacji, które ułatwiają tworzenie zorganizowanego, sprawnie działającego społeczeństwa. Wśród tych metod można wymienić mowę oraz komunikaty niewerbalne przekazywane za pomocą mowy ciała, gestów czy mimiki twarzy. Używane łącznie pozwalają na przekazywanie nie tylko treści, ale i emocji, co można osiągnąć również dzięki sile głosu, jego barwie oraz odpowiedniemu tempu wypowiadania słów. Innym, ważnym sposobem komunikacji jest słowo pisane, które nabiera szczególnego znaczenia dzięki upowszechnionemu dostępowi do sieci Internet. Każda forma komunikacji wymaga nadawcy i odbiorcy oraz odbywa się w pewnym kanale komunikacyjnym. Przesyłane komunikaty zawierają informacje, które powinny być zrozumiałe dla odbiorcy dysponującego wiedzą niezbędną do ich odczytania i zrozumienia. Część wiadomości może być jawna, ale najważniejsze informacje

---

[1] I. Kanter, W. Kinzel, *Neural cryptography, Proceeding of the 9th International Conference on Neural Information Processing*, Singapore 2002.

[2] I. Kanter, W. Kinzel, E. Kanter, *Secure Exchange of information by synchronization of neural networks*, "Europhysics Letters" 57 (2002).

mają charakter poufny i powinny być chronione przed nieautoryzowanym dostępem. Jest to bardzo ważne, gdyż większość komunikacji odbywa się obecnie z wykorzystaniem niezabezpieczonych kanałów komunikacji. Należy się więc liczyć z potencjalnym niebezpieczeństwem podsłuchiwania. W celu zachowania poufności przesyłanych danych stosuje się różne zabezpieczenia kryptograficzne, wśród których najważniejsze jest szyfrowanie komunikacji.

Zgodnie z definicją szyfrowanie jest to przekształcenie tekstu jawnego w szyfrogram w taki sposób, aby oryginalną treść mógł odtworzyć jedynie zaufany odbiorca i by nie była ona dostępna dla potencjalnego atakującego[3]. Współczesne kryptosystemy bazują na problemach trudnych obliczeniowo, a więc takich zagadnieniach matematycznych, dla których nie istnieje albo nie jest znany efektywny algorytm rozwiązania. Dla przykładu jeden z najpopularniejszych kryptosystemów RSA[4] wykorzystuje problem faktoryzacji dużych liczb złożonych, czyli obliczania ich nietrywialnych dzielników. Same metody szyfrowania można podzielić na szyfry ograniczone oraz wykorzystujące klucze kryptograficzne. W systemach ograniczonych tajny jest cały sposób przekształcania treści. Problemem w stosowaniu tego podejścia jest konieczność ustalenia przez zaufane strony sposobu szyfrowania. Obecnie trudno byłoby się spotkać z każdą stroną komunikacji z osobna i ustalić specjalny sposób szyfrowania danych. Trzeba pamiętać, że szyfrowane są wszystkie logowania do kont e-mailowych, kont w serwisach społecznościowych, bankach, gdzie szyfrowane są też polecenia wykonania wszystkich operacji, szyfrowane mogą być również przesyłane treści oraz wiele innych elementów. Aby rozwiązać ten problem, wprowadzono metody szyfrowania wykorzystujące klucze kryptograficzne. Podejście to charakteryzuje się tym, że algorytm szyfrowania jest publicznie znany, ale zależy od dodatkowej wartości zwanej kluczem kryptograficznym. Jest realizacją zasady Kerckhoffsa[5]. W związku z tym całe bezpieczeństwo opiera się na używanych kluczach, czyli na sposobie ich generowania i przechowywania. Istotna jest także matematyczna trudność, jaką stanowi rozwiązanie problemu obliczeniowego, który wykorzystywany jest w stosowanym algorytmie. Generalnie rośnie ona wraz ze zwiększaniem wartości klucza.

Mimo że dla stosowanych w kryptografii algorytmów nie są znane efektywne algorytmy ich łamania, to zawsze pozostają do dyspozycji metody złożone obliczeniowo, a w ostateczności atak brutalny polegający na sprawdzeniu wszystkich możliwych kluczy. Stosowanie gorszych algorytmów wymaga dużo większych mocy obliczeniowych i dłuższego czasu, ale stały rozwój możliwości sprzętu komputerowego sprawia, że taki atak wciąż stanowi aktualne zagrożenie. Wobec tego do zabezpieczenia danych stosuje się coraz dłuższe klucze, co gwarantuje wyższy poziom ochrony, ale skutkuje wydłużeniem czasu potrzebnego na

[3] Zob. A. Menezes, S. Vanstone, P. Van Oorschot, *Handbook of Applied Cryptography*, CRC Press 1996.

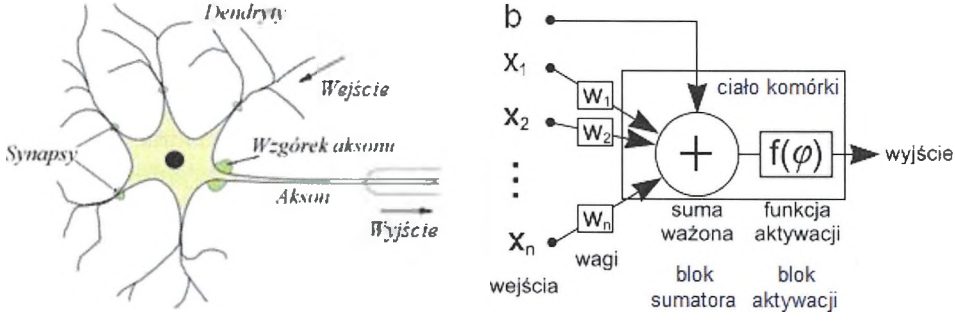
[4] Zob. tamże.

[5] Zob. tamże.

szyfrowanie i deszyfrowanie. Z tych powodów ważnym kierunkiem badań jest poszukiwanie alternatywnych sposobów wykonywania operacji kryptograficznych, które nie opierałyby się na problemach trudnych obliczeniowo. W ten nurt wpisuje się opracowany przez W. Kinzela oraz I. Kantera protokół uzgadniania kluczy kryptograficznych z wykorzystaniem sztucznych sieci neuronowych[6].

Synchronizacja sztucznych sieci neuronowych typu Tree Parity Machine

Sztuczna sieć neuronowa jest matematycznym, znacznie uproszczonym modelem ludzkiego mózgu, który zwykle jest implementowany w programie komputerowym. Najpopularniejszym stosowanym wariantem takiej sieci jest perceptron wielowarstwowy, który zbudowany jest ze sztucznych neuronów ułożonych warstwami. Występująca w naturze komórka nerwowa – neuron – została w uproszczeniu opisana matematycznie w 1943 roku przez McCullocha i Pittsa[7]. Model ten jest jedynie pewną symplifikacją, dlatego nazywa się go sztucznym neuronem. Na rysunku 1 zaprezentowano komórkę nerwową oraz jej schemat.



Rysunek 1. Komórka nerwowa i model McCullocha i Pittsa

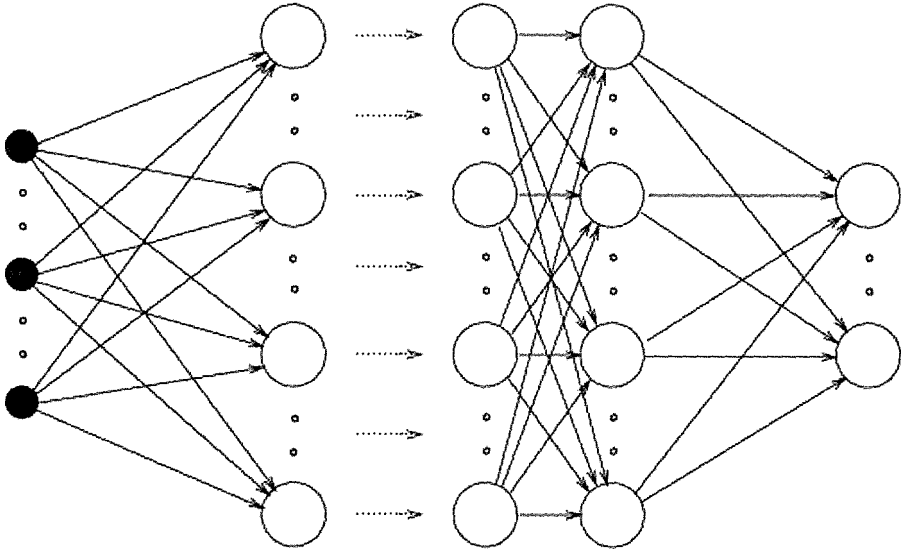
Główne elementy, z których zbudowany jest sztuczny neuron, to blok sumatora i blok aktywacji. Działanie modelu polega na odebraniu w pewnej chwili impulsów wejściowych i przemnożenie ich przez przypisane im wagi. Iloczyny sygnałów i wag zostają następnie zsumowane w bloku sumatora, dając w wyniku sygnał φ określany potencjałem membranowym (postsynaptycznym). W kolejnym kroku jest on przetwarzany przez blok aktywacji, a określona w nim f funkcja przekształca potencjał w sygnał wyjściowy neuronu.

Schemat całej sieci został zaprezentowany na rysunku 2. Neurony pierwszej warstwy przetwarzają sygnały docierające z zewnątrz, a wyniki przekazywane są do kolejnej warstwy i tak dalej, aż do ostatniej warstwy wyjściowej.

---

[6] Zob. I. Kanter, W. Kinzel, *Neural cryptography*...

[7] Zob. W.S. McCulloch, W.H. Pitts, *A logical calculus of ideas immanent in nervous activity*, "Bulletin of Mathematical Biophysics" 5 (1943).



Rysunek 2. Schemat sztucznej sieci neuronowej

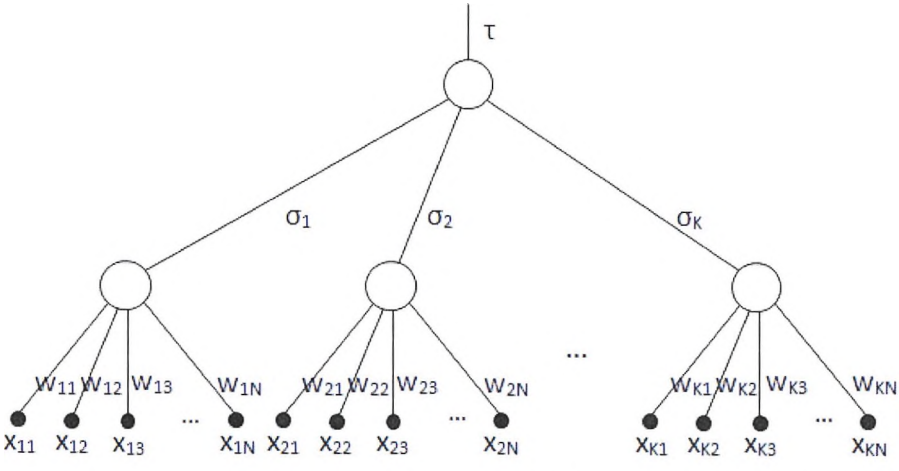
Najważniejszą cechą sztucznych sieci neuronowych jest ich zdolność do uczenia się na podstawie przetwarzanych sygnałów. Odpowiednie algorytmy odpowiadają za modyfikowanie wag sieci tak, aby zwracała jak najdokładniejsze wyniki.

Prezentowany w artykule protokół uzgadniania kluczy kryptograficznych korzysta z charakterystycznych sieci nazywanych *Tree Parity Machine* (drzewiasta maszyna parzystości)[8]. Pierwsza warstwa neuronów tej sieci złożona jest z typowych sztucznych neuronów. W drugiej warstwie znajduje się zawsze tylko jeden specyficzny neuron realizujący operację mnożenia wyników neuronów pierwszej warstwy. W sieci TPM zastosowano rozłączne pola recepcyjne dla każdego neuronu. Oznacza to, że każdy neuron pierwszej warstwy otrzymuje na wejściu własny fragment całego wektora wejściowego. Nadaje to sieci strukturę podobną do *drzewa* (ang. *tree*). Schemat budowy takiej sieci pokazano na rysunku 3.

Topologię sieci TPM opisują dwa parametry: K – ilość neuronów w pierwszej warstwie ukrytej i N – ilość sygnałów wejściowych dla każdego neuronu. Trzecim parametrem opisującym tę sieć jest liczba L, nie określa ona jej topologii, natomiast wskazuje maksymalną (L) i minimalną ($-L$) wartość każdej z wag.

Każdy z impulsów wejściowych przyjmuje wartość -1 albo 1. Jeżeli suma tych sygnałów przemnożonych przez odpowiadające im wagi (potencjał postsynaptyczny) przekroczy 0, to funkcja aktywacji zwróci 1, a w przeciwnym przypadku -1 jako wynik działania danego neuronu pierwszej warstwy. Natomiast neuron z drugiej warstwy oblicza iloczyn docierających do niego sygnałów, a więc liczb

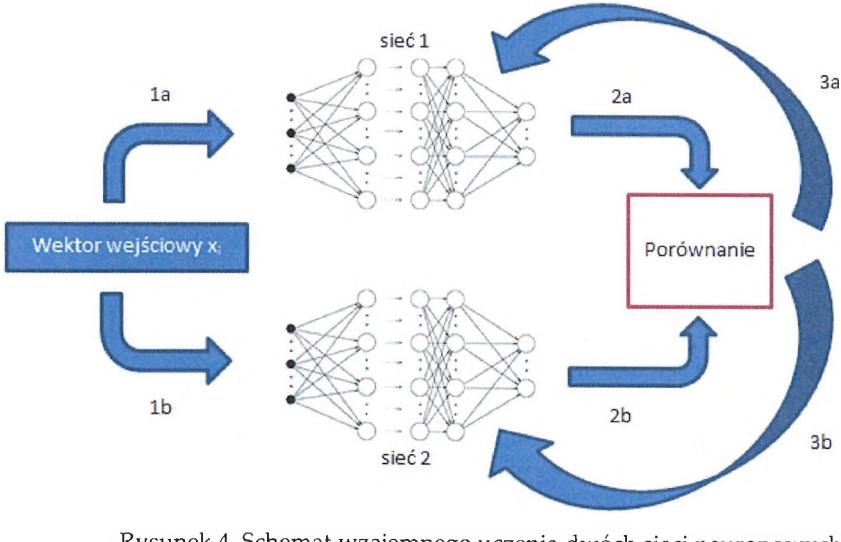
[8] Zob. M. Volkmer, S. Wallner, *Tree parity machine rekeying architectures*, "IEEE Transactions on Computers" 54 (2005).



Rysunek 3. Schemat sieci Tree Parity Machine

1 i -1, gdyż takie są wyniki neuronów warstwy pierwszej. Stąd wynikiem sieci jest też 1 albo -1.

Uczenie sieci TPM odbywa się zgodnie z algorytmem, który jest rozwinięciem metody uczenia sieci z nauczycielem. Jest to przypadek wzajemnego uczenia dwóch sieci, w którym obie sieci uczą się swojego działania nawzajem, będąc zarówno nauczycielem, jak i uczniem[9]. Schemat takiego postępowania pokazany został na rysunku 4. Wagi sieci modyfikowane są zgodnie z jedną z trzech me-



Rysunek 4. Schemat wzajemnego uczenia dwóch sieci neuronowych

[9] Zob. I. Kanter, W. Kinzel, *Neural cryptography*...

tod: *Reguła anty-Hebba, Reguła Hebba, Reguła losowych kroków*[10]. W wyniku takiego uczenia sieci TPM osiągną po pewnym czasie zgodne wartości wag, a stan ten nazywany jest zsynchronizowaniem sieci. Sieci raz zsynchronizowane pozostają w tym stanie bez względu na dalsze ich uczenie i chociaż wagi takich sieci będą się zmieniać, to w obu sieciach w ten sam sposób, pozostając parami równe.

Zjawisko zsynchronizowania sieci może być wykorzystane do zbudowania protokołu uzgadniania kluczy kryptograficznych[11]. Niech A i B będą dwiema stronami, które chcą uzgodnić klucz do szyfrowania dalszej komunikacji. Stosując zaproponowany protokół, wykonują następujące kroki:

0. Strony A i B ustalają, np. przez otwarty kanał, parametry K, N i L opisujące topologię sieci TPM i przedział, do którego będą należały wagi uczonych sieci. Dodatkowo A i B ustalają wspólną metodę ich uczenia.
1. Każda ze stron tworzy własną sieć TPM o losowych, tajnych wagach w^A i w^B.
2. Obie strony otrzymują ten sam, publiczny wektor wejściowy x i obliczają wyniki działania swoich sieci τ^A i τ^B.
3. Strony wymieniają się obliczonymi wynikami działania sieci.
4. Strona A traktuje wynik τ^B otrzymany od B jako oczekiwany wynik działania dla swojej sieci, a strona B analogicznie postępuje z wartością τ^A uzyskaną od A.
5. Obie strony modyfikują wagi sieci zgodnie z wybraną metodą uczenia.
6. Obie strony porównują wyniki działania obu sieci τ^A i τ^B, aby określić ilość kolejno występujących zgodnych wyników. Jeżeli wartość ta jest większa od założonego progu, to należy zakończyć algorytm i uznać sieci za zsynchronizowane, w przeciwnym wypadku należy kontynuować naukę i wrócić do kroku 2.

Po zakończeniu procesu nauki sieci mają zgodne wektory wag, są więc zsynchronizowane. Wagi takich sieci mogą być użyte wprost jako klucze kryptograficzne albo jako punkt startowy dla generatora liczb pseudolosowych, które zostaną użyte jako te klucze. Ciągła zmienność wag w wyniku dalszego uczenia zsynchronizowanych sieci jest dodatkowym utrudnieniem dla potencjalnego atakującego, gdyż musi on nie tylko odkryć jeden klucz użyty do szyfrowania, ale poznać mechanizm jego zmian.

Analiza czasu synchronizacji sieci typu TPM

Proces synchronizacji sieci TPM jest zależny od kilku parametrów: wielkości uczonych sieci, losowo wybranych na początku wartości ich wektorów wag, oraz impulsów wejściowych, losowanych w każdym kroku uczenia. Ze względu na

[10] Zob. tamże.

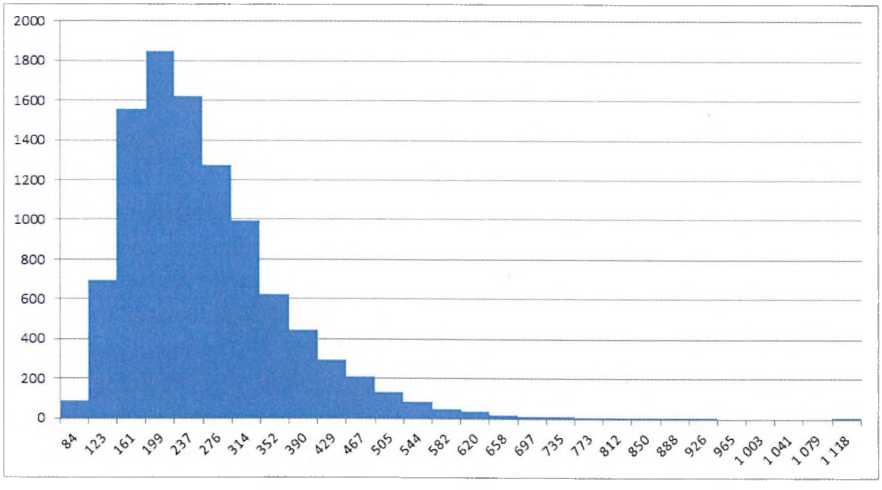
[11] Zob. I. Kanter, W. Kinzel, E. Kanter, *Secure Exchange...*

tę losowość można zaobserwować stosunkowo dużą rozpiętość czasów uczenia sieci o ustalonej topologii. W tabeli 1 przedstawiono czasy liczone ilością kroków uczenia potrzebnych do zsynchronizowania sieci TPM uzyskane po 1000 synchronizacji sieci o różnych parametrach K-N-L. Symulowane sieci uczone były metodą losowych kroków.

Tabela 1. Zestawienie czasów uczenia sieci TPM o parametrach 3-16-L

Parametry sieci	Min czas	Max czas	Średni czas
3-16-1	7	132	33,4
3-16-5	263	2849	755,6
3-16-10	1159	8579	3239,3
3-16-15	2991	20263	7657,9
3-16-20	5044	46057	14444,6
3-16-30	14520	92569	34104,3
3-16-40	23919	161440	64018,9
3-16-50	42856	288219	102382,7

Wraz ze zwiększaniem przedziału wartości wag sieci rośnie również czas potrzebny do osiągnięcia przez TPM zgodnych wektorów wag. Istotnym faktem jest to, że zachowane są charakterystyczne histogramy rozkładu czasów synchronizacji. Wykres 1 przedstawia charakterystyczny histogram dla sieci 3-16-3 uzyskany na podstawie 10000 synchronizacji. Jak pokazano w pracy[12], rozkład ten jest zgodny z rozkładem Poissona.



Wykres 1. Typowy rozkład czasów synchronizacji sieci TPM

[12] M. Dolecki, R. Kozera, *Distribution of the Tree Parity Machine synchronization time*, "Advances in Science and Technology" 18 (2013).

Mimo dużej rozpiętości uzyskanych czasów synchronizacji, można zaobserwować znaczącą liczbę krótkich synchronizacji. Dla analizowanych sieci stwierdzono, że 75% synchronizacji kończy się w czasie będącym około połową najdłuższej obserwowanej synchronizacji[13]. W trakcie wzajemnego uczenia sieci otwartym kanałem komunikacyjnym przesyłane są impulsy wejściowe do sieci oraz wyniki działania tych sieci pobudzonych otrzymanymi impulsami. Każda taka wymiana dostarcza potencjalnemu atakującemu pewnych informacji o sieciach. Występujące rzadziej, dłuższe synchronizacje są z tego powodu potencjalnie bardziej niebezpieczne. Z tego względu istotne jest skupienie się przez strony synchronizujące sieci na krótkich synchronizacjach, które stanowią większość w obserwowanych symulacjach. Nie chcąc kontynuować potencjalnie długich przypadków uczenia sieci, można przerywać toczone synchronizacje, które trwają dłużej niż połowa najdłuższego z zaobserwowanych wcześniej przypadków i nie zakończyły się zsynchronizowaniem sieci. Następnie można rozpoczynać kolejne uczenie sieci z nowo wylosowanymi wagami początkowymi.

Praktyczne wykorzystanie zjawiska synchronizacji sieci neuronowych do konstrukcji protokołu uzgadniania kluczy kryptograficznych wymaga utajnienia wartości wag sieci obu zaufanych stron komunikacji. Okazuje się jednak, że analizując wymieniane między nimi wyniki działania sieci, można ocenić stopień zgodności tych wag bez ich dokładnej znajomości. Pozwala to zaufanym stronom na określenie, czy aktualnie trwające uczenie sieci należy do grupy krótko- czy długotrwałych, ale nie jest to wystarczająca wiedza dla atakującego, który poszukuje dokładnych wartości wag obu sieci. Chcąc określić stopień zgodności wektorów o znanych składowych, można użyć miary euklidesowej, którą dla wektorów wag określono wzorem:

$$dist(A,B) = \|w^A - w^B\| = \sqrt{\sum_{k=1}^{KN} \left(w_k^A - w_k^B\right)^2}.$$

Na początku synchronizacji, gdy wagi są wartościami losowymi, ich odległość euklidesowa przyjmuje duże wartości, które zależą od ustalonego parametru L. W trakcie uczenia sieci ich wektory wag zbliżają się do siebie, co powoduje zmniejszanie tej odległości do zera w stanie zsynchronizowania sieci, gdy wagi mają takie same wartości. W przypadku gdy nie są bezpośrednio znane wartości wag, zaufane strony synchronizacji dysponują jedynie wynikami obu sieci dla wspólnego wektora impulsów wejściowych. W początkowych fazach uczenia zgodne wynik obu sieci przeplatają się z różnymi wynikami ich działania. Podczas uczenia zwiększa się częstotliwość występowania zgodnych wyników obu sieci, a po zsynchronizowaniu, skoro wektory wag obu sieci są zgodne, to i wyni-

[13] Zob. M. Dolecki, *Tree Parity Machine synchronization time – statistical analysis*, "Mathematics, Physics and Informatics Series" 6 (2012), nr 153.

ki dla wspólnego wektora wejściowego są takie same. Tak więc odległość euklidesowa maleje od pewnej wartości początkowej do zera, a częstotliwość rośnie od zera do jedynki. Chcąc porównać obie te wartości, przeprowadzono normalizację i odwrócenie wartości odległości według poniższego wzoru:

$$\mathbf{dist(A,B)}_t = \frac{\max\limits_{1\le j\le t_{synch}} \mathbf{dist(A,B)}_j - \mathbf{dist(A,B)}_t}{\max\limits_{1\le j\le t_{synch}} \mathbf{dist(A,B)}_j - \min\limits_{1\le j\le t_{synch}} \mathbf{dist(A,B)}_j}.$$

Przekształcenie to pozwoliło na określenie współczynnika korelacji oraz błędu średniokwadratowego między częstotliwością występowania zgodnych wyników w określonej liczbie poprzednich kroków oraz odległością wektorów wag[14]. Dla krótkich synchronizacji wartości te wynosiły średnio 0,965 i 0,02 odpowiednio dla współczynnika korelacji i błędu, a dla długich – średnio 0,89 i 0,015. Świadczy to o harmonii analizowanej częstotliwości wymiany zgodnych wyników sieci i odległości wektorów wag. Poniższe tabele prezentują szczegółowe wyniki dla kilku krótkich synchronizacji sieci o parametrach N-K-L równych 3-101-3.

Tabela 2. Współczynnik korelacji pomiędzy odległością wag oraz częstotliwością zgodnych wyników liczoną w określonej liczbie wcześniejszych wymian

	dist, fr 25	dist, fr 50	dist, fr 75	dist, fr 100	dist, fr 125
TPM 1	0,9208	0,9547	0,9726	0,9736	0,9637
TPM 2	0,9773	0,9851	0,9809	0,9758	0,9688
TPM 3	0,9588	0,9820	0,9790	0,9673	0,9508
TPM 4	0,9515	0,9579	0,9652	0,9702	0,9683

Tabela 3. Błąd średniokwadratowy

	dist, fr 25	dist, fr 50	dist, fr 75	dist, fr 100	dist, fr 125
TPM 1	0,0376	0,0188	0,0111	0,0145	0,0256
TPM 2	0,0113	0,0047	0,0074	0,0151	0,0273
TPM 3	0,0089	0,0079	0,0187	0,0368	0,0598
TPM 4	0,0152	0,0114	0,0130	0,0199	0,0319

[14] Zob. M. Dolecki, R. Kozera, K. Lenik, *The evaluation of the TPM synchronization on the basis of their outputs*, "Journal of Achievements in Materials and Manufacturing Engineering" 57 (2013).

Podsumowanie

Czas synchronizacji sieci TPM cechuje się jednak dużą rozpiętością, co ma wpływ na poziom bezpieczeństwa stworzonego z ich wykorzystaniem protokołu uzgadniania kluczy kryptograficznych. Bezpieczeństwo to zależy w istotny sposób od wybranej metody uczenia oraz od wyboru wielkości stosowanych sieci. Nie bez znaczenia pozostaje tu również losowość przy wyborze początkowych wag obu sieci oraz przy generowaniu wektora wejściowego. Szczegółowe analizy czasu synchronizacji sieci pozwalają na wykrycie i przerywanie potencjalnie długotrwałych i mniej bezpiecznych cykli. Zaobserwowana charakterystyka rozkładu czasu synchronizacji uzasadnia sensowność rozpoczynania kolejnej synchronizacji z nowymi wagami. Skupienie się na szybkich synchronizacjach utrudnia zadanie potencjalnemu atakującemu, skracając czas, jaki ma na przeprowadzenie ataku.